UNIVERSITÉ MOHAMMED V
FACULTÉ DES SCIENCES

Numéro d'ordre : 2758

THÈSE DE DOCTORAT

Présentée par

Rkia AOUINATOU

Discipline : Science de l'Ingénieur

Spécialité : Informatique et Télécommunication

LE CHIFFREMENT BASÉ SUR L'IDENTITÉ : IBE, PROPOSITIONS ET MISE EN OEUVRE

Soutenue le 7 Février 2015

Devant le jury composé de :

Président :

M. ABOUTAJDINE Driss : PES, Faculté des Sciences, Rabat

Examinateurs :

M. ZAHID Noureddine : PES, Faculté des Sciences, Rabat

M. NITAJ Abderrahmane : PES, Université de Caen Basse Normandie, France

M. AZIZI Abdelmalek : PES, Faculté des Sciences, Oujda

M. HEDABOU Mustapha : PH, École Nationale des Sciences Appliquées, Safi

M. El HASSOUNI Mohammed : PH, Faculté des Lettres et Sciences Humaines, Rabat

M. BELKASMI Mostafa : PES, École Nationale Supérieure d'Informatique et d'Analyse des Systèmes, Rabat

Laboratoire de **R**echerche **I**nformatique et **T**elecommunication : **LRIT**

Faculté des Sciences, 4 Avenue Ibn Battouta B.P. 1014 RP, Rabat Maroc

Tel +212 (0) 37 77 18 34/35/38, Fax : +212 (0) 37 77 42 61, http ://www.fsr.ac.ma

UNIVERSITÉ MOHAMMED V
FACULTÉ DES SCIENCES

Numéro d'ordre : 2758

THÈSE DE DOCTORAT

Présentée par

Rkia AOUINATOU

Discipline : Science de l'Ingénieur

Spécialité : Informatique et Télécommunication

LE CHIFFREMENT BASÉ SUR L'IDENTITÉ : IBE, PROPOSITIONS ET MISE EN OEUVRE

Soutenue le 7 Février 2015

Devant le jury composé de :

Président :

M. ABOUTAJDINE Driss : PES, Faculté des Sciences, Rabat

Examinateurs :

M. ZAHID Noureddine : PES, Faculté des Sciences, Rabat

M. NITAJ Abderrahmane : PES, Université de Caen Basse Normandie, France

M. AZIZI Abdelmalek : PES, Faculté des Sciences, Oujda

M. HEDABOU Mustapha : PH, École Nationale des Sciences Appliquées, Safi

M. El HASSOUNI Mohammed : PH, Faculté des Lettres et Sciences Humaines, Rabat

M. BELKASMI Mostafa : PES, École Nationale Supérieure d'Informatique et d'Analyse des Systèmes, Rabat

Laboratoire de **R**echerche **I**nformatique et **T**elecommunication : **LRIT**

Faculté des Sciences, 4 Avenue Ibn Battouta B.P. 1014 RP, Rabat Maroc

Tel +212 (0) 37 77 18 34/35/38, Fax : +212 (0) 37 77 42 61, http ://www.fsr.ac.ma

Remerciements

Les travaux présentés dans ce mémoire de thèse ont été effectués à la Faculté des Sciences de Rabat au sein du Laboratoire de Recherche Informatique et Télécommunications (LRIT), sous la direction du Professeur Driss ABOUTAJDINE, responsable du laboratoire LRIT, en partenariat avec Monsieur Mostafa BELKASMI, professeur à l'École Nationale Supérieure d'Informatique et d'Analyse des Systèmes et responsable de l'équipe TSE.

Tout d'abord j'aimerais remercier M. ABOUTAJDINE Driss, Professeur à la Faculté des Sciences de Rabat-Agdal qui m'a fait l'honneur de présider le Jury de cette thèse. Je le remercie pour m'avoir accueilli au sein du Laboratoire LRIT, pour sa disponibilité, sa contribution à ma formation scientifique, son soutien et ses conseils pédagogiques. Qu'il trouve ici toutes mes reconnaissances.

Je voudrais remercier tout particulièrement M. BELKASMI Mostafa, Professeur à l'Ecole Normale Supérieure et d'Analyse des Systèmes (ENSIAS), qui a accompagné mon travail, en me faisant bénéficier de son expérience de chercheur, son encadrement et son écoute favorable lors des séances de travail. Grâce à ses conseils scientifiques, jai également évolué dans le domaine de la rédaction d'articles scientifiques et réussit à exploiter des résultats importants. Je lui suis reconnaissante de tous ses efforts.

Je remercie M. ZAHID Noureddine, Professeur à la Faculté des Science de Rabat-Agdal d'avoir rapporté et juger cette thèse.

Je remercie M. NITAJ Abderrahmane, Professeur à l'Université de Caen Basse Normandie, d'avoir accepté la lourde tache d'être rapporteur et de consacrer de son temps pour juger ce travail.

Je remercie M. AZIZI Abdelmalek, Professeur à l'Université Mohammed Premier-Oujda qui ma fait l'honneur d'être examinateur de cette thèse.

Je remercie M. HEDABOU Mustapha, Professeur Habilité à l'ENSA de Safi, d'avoir accepté d'évaluer cette thèse.

Je remercie M. Mohamed El HASSOUNI, Professeur Habilité à la Faculté des Lettres et des Sciences Humaines de Rabat, pour sa participation au Jury de cette thèse.

Je profite, aussi, de cette occasion pour exprimer mes vifs remerciements à certaines personnes qui ont adhéré de manière indirecte à l'élaboration de cette thèse. Permettez-moi de citer :

— M. AZHARI Abdelhak pour son soutien scientifique pendant ma première année d'inscription et de m'avoir facilité de m'intégrer dans l'équipe LRIT.

— M. RAMRAMI Azzedine pour ces conseils fructueux.

Un grand merci s'adresse à tous ceux qui ont eu la gentillesse de me soutenir de prêt ou de loin. Que tous ceux que je n'ai pas pu involontairement cité me pardonnent, je suis reconnaissante à chacun d'eux.

— Je remercie les membres de l'équipe de LRIT qui sont chaleureux et serviables. J'ai eu un immense plaisir de travailler par moment dans ce Labo. Je remercie, notamment, Amina RADGUI et WARKHANE Mohammed pour leurs soutiens.

— Je remercie, également, l'équipe TSE de l'ENSIAS, qui ont fait preuve de dévouement et d'abnégation. Je remercie, en particulier, Ahlam, Said, Hamid, Mohamed, Ali et Abderrazak.

Au final, je remercie tous les membres de ma famille, ma Mère, mes Soeurs et Frères. C'est avec une grande émotion que je les salue pour leurs soutiens, patience et sacrifices. Je dois avouer qu'aucun mot ne pourrait exprimer ma sincère gratitude pour leurs efforts et encouragements et que sans leur soutien, je n'aurais jamais fait une thèse de Doctorat.

Mes pensés vont aussi vers mes : cousins, cousines, tantes, voisins, voisines pour m'avoir suivi et encourager pendant ces folles années de préparations.

Je suis tellement reconnaissante à ma chère cousine Dr. Badia qui n'a pas hésité à me faire béneficier de son expérience. Je suis également reconnaissante à ma chère cousine Hind pour son soutien.

Au coeur de ma mère

À l'âme de mon père

À Alice et Bob

Résumé

C'est dans les années 1980 où ont brillé les premiers rayons du chiffrement basé sur l'identité (IBE), permettant d'écarter le besoin du service d'assurance pour sécuriser la clé, utilisée par le standard PKI. Le point fort de l'IBE est que la clé publique est sous forme d'identité, bien souvent, ce point offre une flexibilité au bâti des communications sécurisées, couronnées par l'envahissement de l'IBE du domaine réseau, social network et carte à puce.
Divers challenges sont posés devant la réussite du chiffrement basé sur l'identité, on cite : résoudre le problème de Kew Escrow ; permettre l'IBE de rendre les même services tel que la PKI. On note dans ce cadre sa réfute contre les schémas classiques, exemple de RSA ; enrichir son corps ; le conserver contre les attaques physiques ; puis sa mise en oeuvre.
Les travaux de recherche essentiels que nous avons réalisé dans le corps de cette thèse mettent l'accent sur :

— Dans le chapitre 2, on relie l'IBE avec le RSA, on expose en ce sens trois versions, les deux premières développent l'idée de Boneh et al lors d'un projet de DARPA, qui sont basées sur la technique des SEM alors que la troisième est totalement nouvelle dans la littérature, elle conserve le RSA classique.
— Le chapitre 3 est réservé à l'exploitation de nouveaux schémas avec couplages, notre innovation dans ce cadre se traduit en trois schémas : deux (un IBE et l'autre HIBE, nommés respectivement : nouveau IBE schéma et nouveau HIBE schéma) sous le modèle sélective ID et un troisième dans le modèle des Randoms Oracles, appelé quatrième schéma de l'IBE.
— Le chapitre 4 exploite dans un premier temps, une idée issue du protocole de Diffie-Hellman qui permet de résoudre le Key Escrow ainsi que le DoD, dans le deuxième, il expose des méthodes pour réussir et puis défendre l'attaques DPA, une autre idée consiste de prohiber la réussite d'une attaque DFA et cela quel que soit le degré de plongement pair.
— Dans le chapitre 5 on traite la mise en oeuvre de l'IBE, on commence par présenter en détail l'implémentation du quatrième schéma de l'IBE et puis mentionner des résultats concrets une comparaison entre BB1 et le nouveau IBE schéma, utilisant les courbes de Barreto-Naehrig.

Mots clés : *IBE, propositions, performance, RSA, cryptosystèmes avec couplages, Random Oracle, sélective ID, Key Escrow, attaque par canaux caché, implémentation.*

Abstract

The first rays of the Identification-Based Encryption (IBE) have dazzled in front of 1980s, it outweigh the need of the insurance service used by the standard PKI. The highlight of the IBE is that the public key is in the form of identity, this provides flexibility in the secure communications crowned by the invasion of IBE the Network, Social network and Smart Cards.

Various challenges are posed against the success of the IBE, we cite : the problem of Key Escrow ; it's refute against the conventional schemes, such as RSA ; enriching its poor account ; keep its body secure against physical attacks ; then its implementation.

This thesis focuses on the following research :

— In Chapter 2, we connect IBE with RSA, to do so, we develop three versions, the two first develop the idea of Boneh et al in the project DARPA, in which the authors are based on the technique of SEM ; while the third is completely new in the literature, it preserves the principle classic of the RSA.

— Chapter 3 expose new schemes with pairing, they are in global three schemes : an IBE and an HIBE schemes in the model selective ID (named respectively new IBE scheme and new HIBE scheme), the third scheme is in the model Random Oracle, we called it forth scheme for an IBE.

— Chapter 4 examine in a first step the problem of the Key Escrow, using an idea based on a key generated using the protocol of Diffie and Hellman which is also useful to the problem DoD, in the second, it exposes methods to succeed and then defend DPA attacks, another idea prohibit the success of the DFA attack whatever is an even embedding degree.

— In Chapter 5 we treat the implementation of IBE, we detail the implementation of the forth scheme of IBE, then we mention by a concrete results a comparison between BB1 and the new IBE scheme, using the curves of Barreto-Naehrig.

Keywords : *IBE, proposition, performance, RSA, cryptosystems with pairing, Random Oracle, selective ID, Key Escrow, attack with side channel, implementation.*

Table des matières

Liste des abréviations

AES	Advanced Encryption Standard
BB1	First scheme of Boneh and Boyen
BB2	Second scheme of Boneh and Boyen
BBG	Boneh Boyen and Goh
BDHP	Bilinear Diffie-Hellman Problem
BN	Barreto-Naehrig
CA	Certificate Authority
c.à.d	c'est-à-dire
CBE	Certificate Based Encryption
CCA	Chosen Ciphertext Attack
CDHP	Computational DiffieHellman Problem
CL-PKC	Certificateless Public Key Cryptography
CMOS	Complementary Metal Oxide Semiconductor
CPA	Chosen Plaintext Attack
CRL	Certificate Revocation Liste
CSR	Certificate Signing Request
CT	Ciphertext
DARPA	Defense Advanced Research Projects Agency
DBDHP	Decisional Bilinear Diffie-Hellman Problem
DFA	Differential Fault Analysis
Div	Diviseur
Dk-wBDHIP	Decisional k-Weak Bilinear Diffie Hellman Inversion Problem
Dk-BDHIP	Decisional k-Bilinear Diffie Hellman Inversion Problem
DoD	Daniel of Decryption
DPA	Differential Power Attack
D-RSA- SEP	Decision RSA Short Encrypted-Prime Problem
EBDHP	Exponent Bilinear Diffie-Hellman Problem

ECADD	Elliptic Curve Point Addition
ECDBL	Elliptic Curve Doubling
EDLP	Elliptic Discret Logarithm Problem
Exp	Exponentiation
FFS	Function Field Sieve
FFT	Fast Fourier Transformation
FR	Frey, Rück
GMP	Gemplus Multiplication Precision
GPI	General Pairing Inversion problem
GT	Galbraith Test
HIBE	Hierarchical Identification Based Encryption
HODPA	High Order DPA
IBC	Identification Based Cryptography
IBE	Identification Based Encryption
IB-mRSA	IBE with mediat RSA
IBS	Identification Based Signature
IND	Indistinguishability
Inv	Inversion
KDC	Key Distribution Center
LM	Le Monôme dominant
LC	Le Coefficient dominant
LPI	Left Pairing Inversion problem
MDA	Message Digest Algorithm
$\mathbf{M}_F\mathbf{I}$	Miller Inversion Full
MI	Miller Inversion
$\mathbf{M}_L\mathbf{I}$	Miller Inversion Lite
MNT	Miyaji, Nakabayashi and Takano
mod	modulo
MOV	Menezes, Okamoto and Vanstone
mpk	master public key
mRSA	mediat RSA
msk	master secret key

Mul	Multiplication
NFS	Number Field Sieve
NIST	National Institute of Stantards and Technology
NM	Non-Malleability
$NM	Strong Non-Malleability
NTL	Number Theory Library
OAEP	Optimal Asymmetric Encryption Padding
OW	One Way
PDL	Discret Logarithm Problem
PKC	Public Key Cryptography
PKCS	Public Key Cryptographic Standards
PKG	Private Key Generateur
PKI	Public Key Infrastructure
RA	Registration Authority
RPI	Right Pairing Inversion problem
RSA	Rivest Shamir Adelman
SEM	Security MEdiator
SGC-PKC	Self-Generated-Certificate Public Key Cryptography
SHA	Secure Hash Algorithm
sID	selective ID
SPA	Simple Power Analysis
Succ	Success
s^+-ID	selective$^+$ ID
usk	user secret key
XOR	eXclusive OR

Table des figures

Liste des tableaux

Introduction générale

Grâce au développement de l'achat et de l'échange de l'information sur internet, enchère d'une évolution remarquable des réseaux et des cartes à puces et avec le grand succès de l'informatique, on a pu faire sortir la sécurité de sa sphère militaire et diplomatique pour devenir une chose nécessaire et quotidienne. Sécuriser une information est fortement lié aux enjeux d'une science appelée cryptologie "science du secret" ; qui se divise en deux grands axes fondamentaux : cryptographie et cryptanalyse.

La cryptographie est la science de masquage, ce terme tire son origine de deux mots grecs Kruptos et Graphein qui signifient respectivement cacher et écrire. Formellement, elle consiste à ne mettre le message en secret que par les parties mises en jeux. L'échange d'informations par les utilisateurs des réseaux publics en croissance permanente a donné à cette science une importance considérable. On assiste ainsi au développement des méthodes qui masquent l'information et surtout à celles qui garantissent sa fiabilité et la sécurité de ses données.

Une cryptographie plus sûr doit satisfaire quatre propriétés fondamentales :

- Confidentialité : elle permet de s'assurer que l'information ne puisse pas être lue par des personnes non autorisées. Elle peut être réalisée par chiffrement symétrique ou chiffrement asymétrique.
- Authentification : Ensemble des techniques permettant de s'assurer que les données reçues et envoyées proviennent bien des entités légitimes. Les fonctions de hachage, les signatures digitales et certificats figurent parmi ces techniques.
- Intégrité des données : C'est un contrôle de contenu qui permet de vérifier et de suivre la non-altération des données. Nous citons les foncions de hachage comme moyen qui permet de faire ce contrôle.
- Non-répudiation : C'est la façon d'empêcher une entité (émettrice ou réceptrice) de nier la participation dans un échange de données. Les systèmes d'échange avec clé publique et signature numérique assurent la non-répudiation.

Quand au second axe, qui est la cryptanalyse : C'est une science d'attaque qui regroupe tous les moyens pour déchiffrer un texte crypté sans connaître la clé. La cryptanalyse peut se faire de plusieurs manières, à savoir, les attaques : brutes forces, mots probables, paradoxe des anniversaires, analyse fréquentielle ainsi que différentielle. Elles permettent d'attaquer un schéma sous forme symétrique. Dans le cas des schémas asymétriques nous pouvons utiliser d'autres formes, par exemple : un schéma RSA peut être attaqué en réussissant la factorisation de son module. Le schéma El-Gamal ainsi que d'autres, demandent de casser le logarithme discret ou un des problèmes de Diffie Hellman.

Dernièrement, les attaques à canaux auxiliaires ont attiré une grande attention en cryptanalyse. Dans ce type d'attaques, nous analysons les propriétés hardware d'un algorithme où le cryptosystème en question s'est basé. Ce type d'attaques peut être appliqué aux schémas symétriques ainsi qu'asymétriques. Actuellement, suite à l'apparition et à la grande utilisation des cartes à puces, les attaques à canaux cachés ont connu une nette évolution.

Comme il y a la science d'attaque (cryptanalyse), nous trouvons aussi la science qui lui résiste. Au sens de cette dernière les spécialistes essayent soit d'élever le niveau de sécurité des schémas qui demande ceci (RSA, El-Gamal...), ou d'ajouter des contre-mesures (attaque à canaux cachés). Parfois, avant de faire le test des études de simulations, certains attaquants sont posés devant le contrôle afin d'extraire les entités mises en jeu en leur permettant d'accéder aux chiffrés des messages. Pour bloquer ce genre d'attaquants, il faut rendre le schéma anonyme.

Une cryptographie moderne est scindée en deux axes principaux : Systèmes symétriques qui se font via une clé qui doit rester secrète, elle est utilisée pour chiffrer et aussi déchiffrer les messages (comme schématisé dans la figure 1) ; puis les systèmes asymétriques dans lesquels la clé de chiffrement est publique, et la clé de déchiffrement privée (voir la figure 2).

Malheureusement la cryptographie symétrique souffre des anomalies suivantes :

- Problème de transmission de la clé partagée
- Comme il y a beaucoup d'utilisateurs, on trouve le problème de gestion des clé. En effet, pour un nombre n d'entités il est nécessaire de distribuer confidentiellement $\frac{n(n-1)}{2}$ =O(n^2) clés différentes, ce qui est plus complexe.
- Problème de renouvellement des clés.

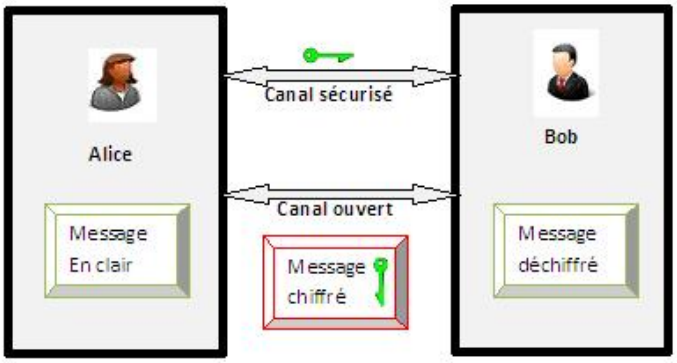


FIGURE 1 – Stratégie de la cryptographie symétrique

En 1976, le monde cryptographie a été bouleversé par la découverte de Diffie et Hellman [63]. Leurs articles donna une nouvelle forme à la cryptographie dite à clé publique (ou chiffrement Asymétrique), et permet d'écarter les problèmes dont souffre la cryptographie symétrique. Le principe de la cryptographie à clé publique (PKC) est simple puisqu'il repose sur l'utilisation de deux clés différentes pour chaque entité ; une publique pour tous le monde, utilisée pour chiffrer le message alors que l'autre reste secrète, elle est utilisée pour déchiffrer le message. La figure suivante illustre son principe.

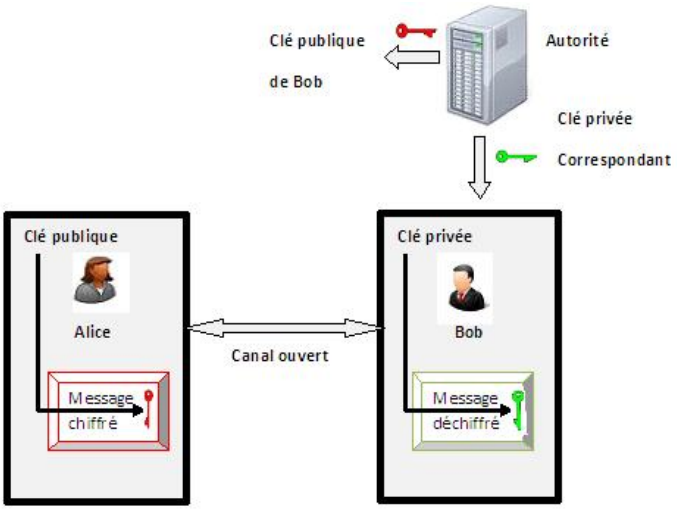


FIGURE 2 – Stratégie de la cryptographie asymétrique

Certes, le concept de la cryptographie à clé publique a été premièrement introduit par Diffie et Hellman, qui étaient incapable de formaliser l'idée de la cryptographie à clé publique par un schéma concret. Le problème reste ouvert jusqu'à ce qu'il est résolu par la proposition des trois cryptographes (Rivest, Shamir et Adleman) selon un schéma qui porte leurs noms, s'appelle le RSA.

Un autre aspect de la cryptographie à clé publique est connu par les schémas de signature.

Ce type de schéma permet de générer une signature sur un message de telle manière qu'elle soit valide par rapport à une clé publique, mais pour que ceci soit vrai, il faut qu'elle se produise en utilisant la clé privée correspondante. Les signatures les plus reconnues sont basées sur : RSA (PKCS6=v2.1 initialisé par Bellare et Rogaway en 1996), schéma de chiffrement à clé publique d'El Gamal (DSA) sans oublier la signature courte proposée en 2001 par Boneh, Lynn et Shacham [21] qui est fondée sur la notion des couplages.

Le chiffrement asymétrique est posé devant le problème de la certification de la clé publique. Chiffrer un message avec un cryptosystème asymétrique requiert la clé publique du destinataire. Le probléme majeur réside dans la légitimité de cette clé publique puisqu'il n'y a rien dans son sujet qui indique son propriétaire légal. Or, il se peut qu'elle tombe facilement de l'attaque du Man in the Middle (l'homme du milieu) en créant des fausses clés publiques après avoir remplacé les initiales. A titre d'exemple, supposons qu'il existe une active attaquante Eve qui rentre dans un système utilisé par les deux communicantes Alice et Bob et qui est capable de modifier les informations de ce système. Eve peut alors se présenter à Alice comme étant Bob et elle fait la même chose devant Bob (se présenter comme étant Alice), elle crée des clés publiques dont elle connaît leur clés privées. Alice qui croit qu'elle utilise la clé publique de Bob chiffre un message à son attention, ce message peut être tout simplement intercepté et lu par Eve, puisqu'en réalité, il est chiffré par sa clé publique.

Par conséquent, il faut faire attention à la clé publique utilisée avant d'envoyer aucun message, en plus, il fallut vérifier que la clé publique obtenue est correcte. Ces deux problèmes poussent à penser à la méthode d'assurance et de certification, connue aussi par PKI.

Un certificat de clé publique est simplement la signature d'une entité de confiance, appelée autorité de certification, en prononçant des affirmations qui argumentent le possesseur de la clé cherchée.

Pour qu'un certificat soit valide, plusieurs conditions doivent satisfaire. Il faut tout d'abord que l'émetteur connaît en avance la clé publique qui lui permet de vérifier le signer. L'autre condition demande l'utilisateur de poser toute sa confiance sur l'autorité de certification et cela sans conditions.

Faire confiance à une entité est un problème très sensible, il soulève plusieurs questions. En faite, qu'est ce qui garantie la confiance rendue à cette entité ? Qu'est ce qui garantie l'origine de la clé publique signée par l'autorité de confiance ? Qu'est ce qui garantie la signature de cette autorité ? Qu'est ce qui garantie les informations stockées dans le certificat généré ? Qu'est ce qui garantie que la clé publique certifiée est bien celle qu'on cherche ? Quest ce qui garantie la durée des certificats désuets ?...

Les autorités de certifications sont sous forme de gouvernement, entreprises ou organisa-

tions. La confiance de la méthode d'assurance PKI peut être admise en utilisant ces autorités, puisque ces dernières peuvent installer un administrateur qui mérite leurs confiances. Tous les utilisateurs qui sont inscrits sous l'une de ses organisations peuvent bénéficier de la confiance rendue à l'administrateur élu. Mais qu'est ce qu'on peut faire dans le cas des communications individuelles ?, les questions posées au dessus demandent ainsi des réponses convenables et définitives.
La méthode de la cryptographie à clé publique sous cadre PKI exige l'obtention en avance de la clé publique du récepteur par la personne qui veux chiffrer un message. Cela nous fait tomber dans la routine des immenses problèmes et qui nécessitent des réponses sur les questions citées ci-après, qu'est ce qu'on peut faire dans le cas où cette clé n'est pas encore préparée ? Au pire des cas, qu'est ce qu'on peut faire si la personne dont on cherche sa clé publique ne peut pas être online ? Qu'est qu'on peut faire si on ne connait que peu d'informations sur cette personne et s'il n'est pas possible de la contacter afin de la pousser à enregistrer elle même chez une autorité ? Qu'est ce qu'on peut faire si cette personne perd sa permission d'inscription dans l'organisation où on peut faire confiance ?...

La plupart des questions posées dans ce manuscrit peuvent être résolues si on utilise le chiffrement basé sur l'identité au lieu de la PKI, connu aussi par ≺ Identification Based Encryption ≻ : (IBE). Dans ce genre de cryptographie à clé publique, la clé publique est sous forme d'une identité. Il suffit alors que l'émetteur connaisse l'identité du récepteur avant même que ce dernier soit inscrit dans un système de communication (son adresse e-mail, son numéro de téléphone, son numéro de CNSS ou tout ce qui peut servir à l'identifier).
Un IBE fonctionne de la manière suivante :

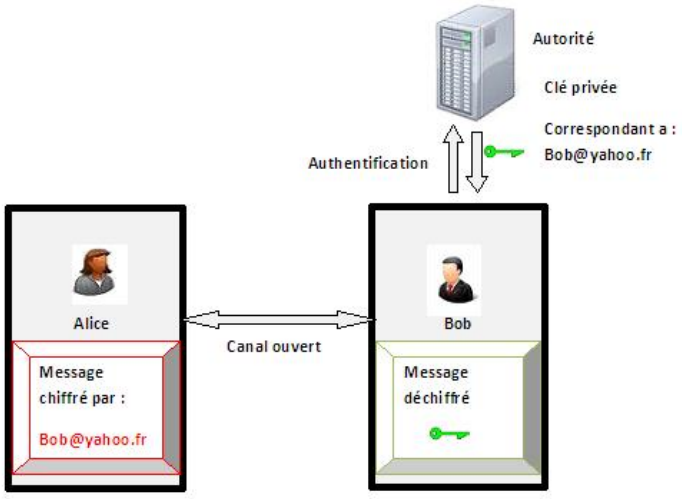


FIGURE 3 – Stratégie du chiffrement basé sur l'identité (IBE)

Dans la figure 3, Alice utilise l'identité de Bob : Bob@yahoo.fr pour chiffrer un message destiné à Bob, le déchiffre de ce message peut se faire en utilisant la clé secrète correspondante à cette identité et cela après avoir bien authentifié Bob.

Utiliser la clé publique sous forme d'identité est sûrement plus pratique que la méthode d'assurance utilisée par le standard PKI. En effet, elle garantit une indépendance lors de la génération de la clé publique, conduit à une flexibilité dans les communications et permet une dispensions des étapes de la génération d'un certificat.
La clé privée dans l'IBE se calcule en fonction de l'identité du récepteur et la clé maîtresse générée par l'autorité responsable de la génération des clés privées, connue par PKG (Private Key Generation). La clé maîtresse se calcule une seule fois et rentre dans toutes les générations des clés privées des utilisateurs qui se diffèrent d'un utilisateur à l'autre suite à l'apparition de l'identité dans leurs corps. Après avoir obtenu un message chiffré à son attention, le récepteur contacte l'autorité responsable de la création des clés privées (PKG) et doit s'authentifier de telle manière à prouver qu'elle est la vraie personne dont le message est chiffré par son identité. S'il ne satisfait pas cette preuve, la personne ne peut nulle part obtenir la clé privée qui lui permet de déchiffrer le message.
L'idée du chiffrement basé sur l'identité (cryptographie basée sur l'identité en général) a été premièrement introduite par Adi Shamir [161] lors de la conférence Crypto84 (en 1984). Il la pose sous forme d'un concept, cependant, l'idée est considérée comme un vrai challenge devant les cryptographes. La réponse à ce défi a nécessité dix sept ans de recherche. C'est en 2001 qu'ont commencé les premières réponses appropriées à cette porte ouverte bien aiguë, il s'agit de deux propositions : celle de Boneh et Franklin [22] et l'autre de Cocks [55].
La sécurité du schéma proposé par Cocks est basée sur le problème du Résidu Quadratique modulo n : trouver r en connaissant r^2 mod n. Le schéma de Cocks est compté non efficace puisqu'il transforme bit par bit, c'est pourquoi l'attention des mondes cryptographes s'accumule vers la proposition de Boneh et Franklin. La raison de son admission réside au fait que ce schéma utilise une notion mathématique s'appelant couplage, l'avantage que porte ce dernier est qu'il a comme propriété la bilinéarité, celle ci rend le chiffrement et le déchiffrement plus simples.

$$e(sP_{pub}, tP_{ID}) = e(tP_{pub}, sP_{ID}) = e(tP_{pub}, P_{priv}) \quad (1)$$

Le P_{pub} et sP_{pub} sont des paramètres publiés à tout le monde.
s est la clé maîtresse, P_{ID} paramètre lié a l'identité, il est construit en utilisant un paramètre public P et l'identité ID.
$P_{priv} = sP_{ID}$ représente la clé privée

t est un paramètre choisit par celui qui émis (chiffre) le message et e le couplage
Bien entendu, le message m peut être chiffré par m $\bigoplus$ e(sP_{pub}, tP_{ID})=c et il peut être déchiffrer par c $\bigoplus$ e(tP_{pub}, sP_{ID}).

Cadres et sujet de la thèse

Charger une autorité de construire les clés privées des utilisateurs pose le problème du Key Escrow qui permet à la PKG de lire les messages des utilisateurs. Si on construit la clé privée d'un utilisateur, on est tous les moyens qui permettent de déchiffrer son message ou de forger sa signature.
En effet, dans certains cas, le Key Escrow peut rendre des services bien utiles ; il peut être utilisé pour récupérer les clés privées perdues, comme il peut permettre aux puissantes autorités tel que le gouvernement de suivre certains utilisateurs dont on doute de leurs comportements. Mais, dans la plupart des cas le Key Escrow est non désirable et il serait réferable de rendre les communications des utilisateurs indépendantes de toute suivie extérieure.
Par ailleurs, depuis longtemps, on se demande si le chiffrement basé sur l'identité peut rendre les mêmes services tels que la PKI, plus particulièrement, est ce qu'on peut se baser sur l'IBE au lieu de la PKI lors de l'utilisation des schémas classiques qui bénéficiaient de la PKI comme le RSA, l'échange des clés de Diffie Hellman et El Gamal.
A part la direction de relier l'IBE avec les schémas classiques, il est usuel que la plupart des propositions des schémas de chiffrement basé sur l'identité sont basés sur les couplages. Si on examine les schémas dédiés en ce sens, on remarque qu'ils sont peu nombreux, alors, on se demande si on ne peut pas opter pour d'autres possibilités.
Ainsi, la réussite d'un IBE revient au pourcentage de son envahissement des outils informatiques, électroniques et réseaux. Cela ne peut se faire qu'après avoir réussi son implémentation. La question qui se pose, est-ce qu'il est toujours possible de traduire ce qui est proposé pour l'IBE sous forme théorique par des résultats concrets et pratiques ?
Implémenter les schémas des IBE dans les outils électroniques tels que les cartes à puce instance d'étudier leurs sécurités devant les attaques physiques telles que celles par canaux cachées.
Enfin, il est temps de standardiser l'IBE comme technologie qui remplacera un jour la PKI.

Notre contribution

A travers cette thèse, on essaye d'apporter quelques réponses aux grandes questions déjà évoquées. Notre contribution en ce sens est triple. Il s'agit pour nous de :

— Combler quelques lacunes de l'IBE

— Ajouter des branches dans sa construction,

— Relater quelques implémentations.

Ainsi, concernant les lacunes, nous nous somme intéressé dans un premier temps au problème de Key Escrow. Pour gérer ce problème, nous avons proposé une méthode basée sur l'utilisation d'une clé de Diffie-Hellman et nous avons montré qu'elle résiste aussi au problème DoD en faisant des vérifications avant de calculer la clé résultante. Ensuite, nous avons traité les attaques par canaux auxiliaires, plus particulièrement, nous avons proposé des méthodes pour réussir et puis défendre l'attaque DPA face au couplage. Pour réussir l'attaque, nous avons traité les deux cas : secret dans le premier et dans le second argument du couplage, ensuite, nous avons traduis nos propositions (pour les deux cas) sur la cryptographie IBC. Pour défendre l'attaque DPA, nous avons examiné les traditionnelles contre-mesures, puis proposé des alternatives convenables. Une autre idée consiste de défendre la réussite d'une attaque DFA et cela quelque soit le degré de plongement pair.

Dans le cadre de sa construction, nos contributions sont de deux types : cryptosystèmes sans couplage et avec couplage. Pour ceux sans couplage nous avons examiné l'anonymat du schéma de Cocks en proposant un schéma qui garde la même longueur que celui de Cocks et garantit un anonymat efficace ; puis nous avons lié le RSA avec l'IBE, dans ce cadre, nous avons proposé trois versions.

— Les deux premières développent des idées de Boneh, D., Ding, X. et Tsudik, G qui utilisent une autorité médiate appelée SEM et le fait de diviser la clé privée entre cette entité et l'utilisateur.

— La troisième version est totalement nouvelle dans la littérature (on ne peut pas utiliser le SEM), elle conserve le principe du RSA classique.

Quand aux cryptosystèmes avec coupalge, nous avons proposé trois schémas :

— Deux schémas sous forme IBE et HIBE dans le modèle sélective ID, appelés respectivement nouveau schéma IBE et nouveau schéma HIBE,

— Un troisième dans le modèle des Random Oracle, qu'on le nomme quatrième schéma de l'IBE.

Enfin et avant de traiter l'implémentation des IBE, nous présentons une attaque qui permet de briser CDHP et BDHP en utilisant l'outil couplage de Tate. Nous donnons alors les conditions à tenir en compte dans les schémas des IBE pour défendre cette attaque. Ensuite, nous détaillons l'implémentation de la version basique du quatrième schéma sous le Random Oracle en utilisant les paramètres : niveau de sécurité 64-bits, courbes MNT où k= 3 et un discriminant D=19. Aussi nous traitons mais sans détail l'implémentation de BB1 et le nouveau schéma IBE dans le niveau de sécurité 255-bits et en évoquant d'autres implémentations.

Plan de la thèse

Le développent de cette recherche est axé à cinq chapitres :

— Dans le chapitre 1, nous présentons un état de l'art sur l'IBE en rappelant ces principales caractéristiques

— Ensuite, le chapitre 2 consiste à examiner les cryptosystèmes de l'IBE sans couplage en apportant un anonymat efficace au schéma de Cocks et en reliant l'IBE avec le célèbre schéma RSA.

— Le chapitre 3 est reservé, quand à lui, aux schémas avec couplage, notre innovation en ce sens se résume en trois schémas.

— Par ailleurs, le chapitre 4 a pour objectif de pallier quelques lacunes de l'IBE, il s'agit du problème de Key Escrow et les attaques par canaux cachés.

— Finalement, le chapitre 5 est consacré à l'examen de la sécurité de l'IBE et à sa mise en évidence.

Chapitre 1

Généralités sur l'IBE

Le présent chapitre est dédié à un état de l'art vital qui référencie le parcourt de la thèse, il consiste à définir le chiffrement basé sur l'identité (IBE), objet de notre recherche. On estime qu'un rappel au sujet de la PKI est nécessaire avant de traiter ce qui différencie cette ancienne technologie de l'IBE.
Ensuite, on donne une base mathématique fondamentale selon laquelle l'IBE s'est basée.
Par ailleurs, l'état de l'art de l'IBE est un axe introductif portant un éclairage sur les avancées de cette technologie. Une partie consacrée au principe de l'IBE mettra en évidence les problèmes rencontrés et ses périmètres.
Enfin, on rappelle, les manières des études de simulations qui permettent de sécuriser l'IBE, relations entre quelques problèmes de Diffie-Hellman utilisés par certains schémas de la cryptographie basée sur l'identité, plus spécialement l'IBE.

1.1 Rappels sur le Publique key Infrastructure

Dans la cryptographie asymétrique, l'échange des clés publiques se fait à travers des lignes non nécessairement sécurisés, ce qui requiert leur authentification avant de les utiliser. Pour cela, le NIST (National Institute of Stantards and Technology) décide de poser un standard dont le but est de mettre à disposition la tâche d'authentification, ce standard a vu le jour en 1994, il est connu par Publique Key Infrastructure (PKI).
Un PKI est un ensemble de moyens combinés par des politiques, mis en oeuvre par des personnes échéantes avec une confiance visée. La PKI permet de générer les clés pour une utilisation cryptographique, de les distribuer, de les conserver, de produire des certificats et de publier les certificats désuets nommés CRL (Certificate Revocation Liste) suivant un annuaire.
La clé publique peut être alors garantie en utilisant une autorité responsable sur la PKI.

Le premier but derrière une PKI est d'établir la confiance dans les échanges entre plusieurs personnes. Avant l'échange, elle garantit l'authentification des sociétaires. Pendant l'échange, elle garantit la confidentialité et l'intégrité des messages. Après l'échange, elle garantit la non répudiation.

1.1.1 Acteurs d'un PKI

La procédure humaine qui commande la PKI est la suivante :

- Autorité d'enregistrement RA (Registration Authority)
- Autorité de certification CA (Certificate Authority)
- Entités Finales

Autorité d'enregistrement (AR ou RA (Register Authority))

Cette autorité s'occupe de vérifier le contenu du CSR (Certificate Signing Request) de chaque nouveau utilisateur dans la PKI. Il s'agit généralement de vérifier l'identité du nouveau utilisateur, son organisation, la relation qui le relie avec la personne dont il a demandé la clé publique...
Les méthodes de vérification peuvent commencer par un simple courier électronique pour finir par des demandes qui contiennent plus d'informations.
Après cette première étape de la procédure, le nouveau utilisateur doit s'adresser à l'autorité de certification.

Autorité de certification (AC ou CA (Certified Authority))

C'est une autorité de confiance qui sert à la création du certificat. Cette dernière consiste à lier l'identité d'un utilisateur à sa clé publique. La AC signe le certificat créé à l'aide de sa clé privée.
La clé privée de la AC doit être protégée contre les réseaux malveillants, c'est pourquoi elle ne se connecte pas à Internet et doit être isolée complètement de tous ce qui est réseau.
Pour certaines applications, la AC peut prendre à sa charge la totalité des opérations de la AR. Mais son vrai rôle consiste à mettre à jour les certificats qu'elle a généré.
Ainsi, elle intervient pour modifier les informations liées au AC quand l'utilisateur change d'activité comme elle peut révoquer le certificat même avant la date de son expiration quand elle estime que l'utilisateur n'ait pas de bonne foi.
Par la suite la AC transmet la liste des certificats révoqués à la CRL qui est soumis à son contrôle.

Normalement il n y a pas de règles nettes qui contrôlent le certificat délivré par la AC, la confiance se fait sottement par transitivité :

— A fait confiance à AC

— La AC délivre un certificat à B

— A est assuré de l'identité de B

Entités Finales

Il en existe deux : Service de publication et Porteur du certificat

- **Service de publication :** Le service de publication est une composante qui rend disponible les certificats des clés publiques émises par une AC à l'ensemble des utilisateurs de ces certificats. Il peut être disposé en ligne, son rôle est de contrôler en temps réel la présence ou pas d'un certificat dans la liste de révocation. La publication peut se faire, soit par annuaire, par serveur Web ou par main en main.
- **Porteur du certificat :** C'est l'entité qui figure dans le certificat et qui possède aussi bien la clé publique que la clé privée qui lui correspond. C'est lui qui active le mécanisme de création de signature et qui possède les données d'activation de sa fonction de signature.

1.1.2 Types de répartitions des CA

Un seul CA ne suffit pas pour répondre à toutes les demandes à travers l'univers, il est donc indispensable d'utiliser plusieurs CA répartis dans diverses régions. Trois types peuvent être envisagés dans la répartition des CA : sous forme hiérarchique, peer to peer, en pont.
Le type hiérarchie se base sur un CA root (directeur) qui possède un certificat autosigné (signé par sa propre clé), en dessous il existe des sous CA appelés subordonnées. Ces derniers reposent sur une chaine de confiance entre eux et le CA root. Pour chaque demande les CA subordonnées envoient leur clés publiques aux CA root qui leur génére le certificat demandé.
Le type CA peer to peer suit la même méthode que le type hiérarchie, sauf que dans ce cas on n'a pas de CA root. Un ou plusieurs CA s'échangent mutuellement leurs clés publiques afin de générer des certificats de manière croisée, les certificats générés se nomment co-signé ou co-certifié.
Ces deux types ne sont pas assez pratiques. En effet, le type hiérarchie nécessite avoir un CA root où il faut stocker en prudence sa clé privée dans un endroit sûr et répondre à toutes les demandes. Or, il sera difficile de trouver ce CA root qui peut prendre cette responsabilité. Le type peer to peer, quand à lui présente la difficulté de croiser le service pour un nombre plus

grand des CA. Par conséquent, le type pont a été créé en vue de surmonter les faiblesses de peer to peer et hiérarchie. Dans ce dernier un CA bridge est mis au service, deux CA échangent leurs clés publiques avec celui ci.

1.1.3 Validation d'un certificat

Avec le grand besoin à utiliser un Certificat, plusieurs normes sont proposées, le plus communiqué est celui de X509.

Méthode utilisée pour obtenir un certificat

Le fonctionnement d'une PKI se résume ainsi :

- *Enregistrer et vérifier les demandes de certificats, étape faite par une Autorité d'enregistrement.*
- *Créer et distribuer des certificats, étape déroulée par une Autorité de certification.*
- *Vérification de validité des certificats, étape faite par une Autorité de validation*
- *Gérer à tout moment l'état des certificats et prendre en compte leur révocation en renouvelant le Dépôt des listes de certificats révoqués-CRL (Certificate Revocation List).*
- *Publier les certificats dans un endroit appelé Dépôt de certificats (Annuaire).*

La méthode utilisée pour obtenir un certificat est bien résumée dans le schéma ci-dessous.

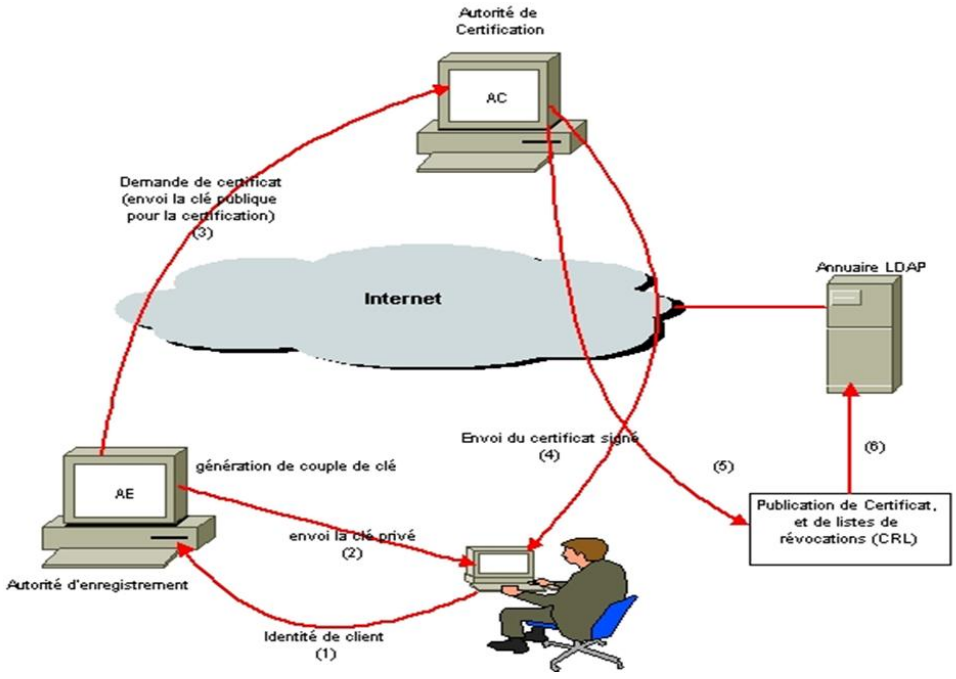


FIGURE 1.1 – Méthode utilisée pour obtenir un certificat

Pour obtenir un certificat le porteur Bob enregistre lui même chez l'AR, après avoir authentifié Bob, cette autorité envoie une clé privée (de signature) à Bob et en même temps elle demande le Certificat de AC en lui donnant toutes les informations sur Bob. Après avoir calculé le certificat, l'AC l'envoi à Bob, ensuite elle la publie suivant un annuaire.

Pour être sûr du contenu d'un certificat, il est recommandé de faire une chaîne de vérifications, résumée ainsi :

Signature de AC	Vérifier que le certificat est authentique et intégre.
Chemin de certification	Vérifier que le service de certificat conduit à un AC confiant.
Période de validité	Vérifier que le certificat mis en jeu n'est pas expiré.
Statut de certificat	Vérifier que le certificat n'est pas révoqué.

1.1.4 Service de révocation CRL

Lorsq'un utilisateur change sa place ou s'il perd sa clé privée, un service informe que ceci doit être mis en place par la PKI. Il est impossible de retirer un certificat révoqué puisqu'il parait chez plusieurs AC. Pour contourner ce problème, un service de révocation est mis en place par le AC, il s'appelle CRL (Certificat Revocation Liste). Pour chaque révocation, le AC enregistre la demande de révocation et vérifie son authenticité. Une fois cette vérification effectuée, la liste des certificats révoqués (CRL) doit être publiée. Cette liste de révocation de la PKI doit disposer d'un service d'annuaire obligatoirement connecté à Internet.

1.1.5 PKI, choses utiles et inutiles

Choses utiles

- Être sûr des clés publiques utilisées
- Garantir la sûreté des clés publiques stockées
- Vérifications ouvertes à l'autorité et à l'utilisateur
- Flexible à n'importe quelle catégorie d'utilisateurs
- Ouvert à tout le monde sans exception
- Permission aux utilisateurs de générer leurs clés personnelles

Choses inutiles

- Système lourd
- Mise à jour quotidienne
- Frais d'enregistrement trop chèrs

- Le récepteur doit laisser sa clé publique chez l'autorité avant d'être contacté
- Suivi continu de la part des utilisateurs ainsi que de la part autorité
- Recherche continue sur les informations qui concernent les utilisateurs inscrits
- Accès complexe au système (beaucoup d'étapes à suivre)
- Authentification difficile
- Clés longues, elles génèrent une complexité importante

1.1.6 Schéma RSA

Parmi les schémas qui bénéficient des services de la PKI nous prenons le RSA. Le RSA est un algorithme introduit par Rivest, Shamir et Adleman en 1978 dans [151] (il leur permettra d'accrocher le Prix Turing de l'ACM en 2002 qui est équivalent au prix Nobel dans le cadre informatique). Pour créer ses paramètres publics Bob (considéré comme récepteur) calcule deux entiers n_b (module) et e_b (exposant). Avec n_b = $p_b \times q_b$ est le produit de deux grands nombres premiers p_b et q_b, e_b est tel que $3 \leq e_b \leq n_b$ et pgcd(e_b,$\varphi(n_b)$) = 1 où $\varphi(n_b)$ = (p_b - 1)(q_b - 1) appelé fonction indicatrice d'Euler. Bob publie ces paramètres via la PKI c.à.d. en utilisant le certificat. Il calcule également l'entier d_b tel que : $0 < d_b \leq n_b$ et $e_b d_b$ = 1 mod $\varphi(n_b)$. Le d_b est considéré comme clé privée connue seulement par Bob. Lorsque Alice veut envoyer un message m $\grave{a}$ Bob, elle va chercher le couple (n_b,e_b) et calcule c = m^{e_b} mod n_b qu'elle envoie comme message chiffré. Bob qui a la clé privée d_b déchiffre simplement c.
Il se peut que l'autorité calcule les paramètres publics n_b et e_b pour Bob. Dans ce cas, Bob est demandé de recevoir la clé privée à travers un canal sûr, mais l'autorité suit tous les messages envoyés à Bob. Alors, il sera mieux d'utiliser la première façon.

Exactitude du crypto système RSA

Extraire m à partir du chiffré m^{ed} mod n parait flou. Mais on a :

$$ed = 1 \ mod \ \varphi(n) \ = \ 1 \ mod \ (p-1)(q-1) \ (pour \ un \ n = pq) \tag{1.1}$$

Alors

$$ed = 1 + k(p-1)(q-1) \tag{1.2}$$

Donc

$$m^{ed} = m^{1+k(p-1)(q-1)} \tag{1.3}$$

On a $m^{p-1} = 1$ mod p de même $m^{q-1} = 1$ mod q (en utilisant le théorème de Fermat)
Le théorème du reste chinois implique que

$$m^{(p-1)(q-1)} = 1 \ mod \ pq = \ 1 \ mod \ n \tag{1.4}$$

Au final on trouve que :

$$m^{ed} \ = \ m^{1+k(p-1)(q-1)} \ = \ mm^{k(p-1)(q-1)} \ = \ m(1 \ mod \ n)^k \ = \ m \ mod \ n \tag{1.5}$$

□

1.2 Le chiffrement basé sur l'identité(IBE)

Du fait des problèmes figurés dans l'utilisation d'un certificat (introduction, partie 1.1.5, nous renvoyons aussi à [72] pour une culture plus approfondie), Adi Shamir cherchait un système pour les crypto systèmes (schéma en général) à clé publique où les certificats sont simple et même non utilisés. La première idée posée à Shamir est de chercher un système où l'utilisateur pourrait fixer lui-même sa clé publique comme une chaîne de caractères et pourquoi pas être liée à l'identité. Shamir présente son idée [161] pendant la conférence CRYPTO qui s'est déroulée en 1984, mais sous forme de concept et il a restreint à traduire son idée uniquement sur les schémas de signatures. Ici, nous présentons sa proposition.

Description de la signature de Shamir

Clé Secrète : g (secret) sachant que : i = g^e (mod n), i est l'identité.

S : s=g.$r^{f(t,m)}$, r est choisi arbitrairement par l'utilisateur tandis que f est choisi par la PKG.

t : t=r^e

On envoi (s,t) et on vérifie si :

$$s^e \ = \ i.t^{f(t,m)} \ mod \ n. \tag{1.6}$$

Après l'idée de Shamir, la clé publique peut être une chaine de caractères liés à l'identité de l'utilisateur, ça peut être son adresse email, son numéro de téléphone, son numéro de CNSS, en général tous ce qui caractérise l'identité. Cette dernière doit être définie de manière unique et que chaque individu ait son identité numérique propre.

Cas d'un ID sous forme d'une adresse email

Si Alice veut envoyer un e-mail à Bob elle cherche uniquement son identité, par exemple elle utilise son adresse email Bob@yahoo.com. Alice peut chiffrer son message en utilisant comme clé publique Bob@yahoo.com. Elle n'a pas besoin donc d'obtenir cette clé publique via un système de certificats. Quand Bob reçoit le message chiffré, il s'identifie auprès d'une autorité de confiance appelée Générateur de clés Privées (PKG) et obtient sa clé privée construite à partir d'une clé maîtresse. Avec cette affaire, Alice peut envoyer son message chiffré à Bob avant même qu'il n'a ni clé privée, ni moyen de le déchiffrer. Bob peut demander cela lorsqu'il reçoit la requête représentée sous forme de message chiffré. Mais ce système de fonctionnement donne à l'autorité PKG une immense validité de suivre ses clients puisqu'elle génère leurs clés privées.

Restriction d'utilisation d'une identité et solution d'attributs

Dans l'IBE, chaque utilisateur est identifié par une seule identité. À priori l'utilisateur ne peut pas être identifié d'une seule manière. Donc les régisseurs d'IBE ont élus l'utilisation des attributs. De cette manière l'utilisateur est caractérisé par un ensemble d'affirmations qui peuvent être son rôle (fonction), son classement ou hiérarchie (expert ou personnage moyen), son autorisation (tâches auxquelles il a accès), son âge plus son identité. Shai et Waters sont les premiers [153] qui ont décrit un schéma avec lequel l'utilisateur peut envoyer un message en spécifiant un ensemble d'attributs et un nombre d. Alors le récepteur qui vérifie au moins d attributs peut déchiffrer le message.

L'IBE et l'Hiérarchie

Comme dans la PKI (partie 1.1.2), nous avons besoin d'une hiérarchie pour éviter la concentration sur une autorité. Les articles [104][90] ont pris l'initiative de relier l'IBE à cette technique, en créant ce qu'on appelle dans la littérature HIBE.
Dans le HIBE on suit une méthode successive où une autorité cerveau, nommée root placée à la tête de la hiérarchie et prenant le rôle du directeur d'une société génère la clé maîtresse à ses subordonnées (sous autorités) classées selon des niveaux. Ainsi, l'autorité du niveau 1 procrée des clés à son sous autorité c.à.d. de niveau 2 et ainsi de suite. De ce fait, l'utilisateur n'est pas obligé de contacter l'autorité root (racine), mais suivant sa résidence il peut joindre l'autorité la plus proche de lui. De cette manière, on simplifie la tâche pour l'utilisateur et aussi pour l'autorité puisqu'il y a des autorités coopératives.

Afin de bien s'adapter avec le fonctionnement de HIBE, nous renvoyons le lecteur à l'article [90] où le schéma de Boneh et Franklin a été implémenté comme exemple.

1.2.1 Les problèmes majeurs de sécurité rencontrés dans l'IBE et les solutions proposées

Stockage de la clé maîtresse (master)

La première difficulté affrontée par les créateurs de l'IBE est de garantir un endroit sécurisé pour stocker la clé maîtresse. Si un attaquant réussi à attaquer ou plutôt connaître la clé maîtresse, il pourra signer n'importe quel message comme étant la PKG, ou encore déchiffrer les messages des utilisateurs en génèrant ses propres clés privées. Pour défendre ceci, des solutions suggèrent de distribuer la clé maîtresse entre plusieurs PKGs.

Key Eskrow

L'IBE connait un deuxième inconvénient souvent décrit comme fondamental, le fait qu'une PKG peut générer les clés privées, permet de lire les messages de ses utilisateurs et donc suivre les communications en court. La solution est beaucoup plus critique lorsqu'il s'agit des secrets sensibles. Nous allons bien traiter ce problème et les solutions offertes dans le chapitre 4.

Séquestre des clés

Dans l'IBE, la PKG génère les clés privées en assurant automatiquement un séquestre de ces clés puisqu'elle peut générer à tout moment une clé privée. Ce qui peut présenter l'avantage de récupérer les clés perdues et constituer des nouvelles, par contre la PKG peut contrefaire une signature et donc déséquilibrer le principe de non répudiation qui est le but fondamentale dans la cryptographie. Parmi les solutions proposées pour remédier à ce problème est de distribuer les clés privées sur plusieurs PKGs possédant chacune sa propre clé maîtresse et récupérer ces clés en cas de besoin.

Révocation des clés

Concernant la PKI, la révocation des clés est garantie par une publication de CRL (Certificate Revocation List). Ce qui n'est pas le cas dans l'IBE, puisqu'on n'utilise ni un certificat ni un annuaire où on peut stocker les clés révoquées. Donc, lorsqu'une clé privée est compromise ou qu'un utilisateur a changé sa place, il faut le confirmer. Boneh et Franklin [22] proposent d'utiliser l'identité concaténée avec une chaîne de caractères d'horodatage. Par

exemple, au lieu d'utiliser l'adresse email de Bob : Bob@yahoo.fr, il vaut mieux le remplacer par Bob@yahoo.fr||05||2014. Malheureusement cette méthode demande de renouveler les clés privées à chaque fois, une méthode plus adéquate a été suggérée par Alexandra Boldyreva et al [18].

Transmission de la clé privée

Pour ce qui est de la PKI, la clé privée est générée par l'utilisateur lui même après sa publication par l'autorité compétente. Ce qui lui permettra de se défendre suite à une éventuelle attaque du Man in the Middle. Dans le cadre de l'IBE, c'est la PKG qui génère la clé privée et pour la transmettre aux utilisateurs, ça demande un canal sûr ou de se présenter physiquement pour l'obtenir. Pour décroitre ce besoin une solution a été proposée par Kumar en 2006 [124], nous rappelons ici sa méthode dans le cas de la clé privé utilisée par le schéma de Boneh et Franklin :
Tout d'abord, l'utilisateur qui a l'identité ID, pour s'authentifier auprès de l'autorité PKG, doit choisir un r_{ID} (dans un corps fini convenable) et il l'envoie à cette dernière sous la forme : $<$ ID,(r_{ID}^{-1} mod q)P $>$. La PKG stocke quelques part le (r_{ID}^{-1} mod q)P comme paramètre réservé à ID et il issu après sQ_{ID}||(r_{ID}^{-1} mod q)P comme preuve d'enregistrement à l'utilisateur.
Pour renvoyer la clé privée, la PKG choisit x dans le corps généré, il calcule ensuite U=sQ_{ID}+xP, V=x(r_{ID}^{-1} mod q)P. Une fois reçu <U,V>, l'utilisateur concerné c.à.d. celui qui a généré r_{ID} peut extraire facilement la clé cherchée en calculant $U - r_{ID}V = sQ_{ID}$.

La même méthode peut être appliquée dans le cas des autres schémas.

1.2.2 Points en communs et différences entre l'IBE et la PKI

Le tableau 1.1 présente les points permettant de comparer entre l'IBE et la PKI, selon non seulement notre point de vue ; mais aussi d'après les regards de certains chercheurs, en se basant sur [93][149][73].

$\surd$: Veut dire que cela existe
$\frac{\surd}{2}$: Veut dire moyenne existence
$\times$: Veut dire n'existe pas.

On exprime par genre (inconvénient/ avantage) important lorsqu'il présente une difficulté (sa solution est ardue) ou une flexibilité d'utilisation. Par contre, l'utilisation et la solution d'un peu important est à peu près simple (ça dépend du problème qu'il présente).

Points cités/ genre	inconvénient		avantage		important	peu important
Points cités/ technologie	PKI	IBE	PKI	IBE		
Demande de certificat	√	×			oui	
Besoin la confiance de l'autorité	$\frac{\surd}{2}$	√			oui	
Key escrow	×	√			oui	
Authentifier le récepteur et l'émetteur	√	×				oui
Chercher une forme spéciale des clés vérifiant certaines critères	√	×				oui
Taille plus grande de la clé publique	√	×			oui	
Demande CRL ou OCSP			√	×		oui
Authentification de haut niveau	√	×				oui
Le récepteur laisse sa clé publique chez l'autorité avant d'être contacter	√	×				oui
Besoin d'un stockage sécurisé d'une clé fondatrice du système (clé master)	×	√			oui	
Permet un off line de chiffrement			×	√		oui
Permet aux utilisateurs de générer leurs clés personnelles			√	×		oui
Produire la clé publique à partir de l'identité			×	√	oui	

TABLE 1.1 – Comparaison entre l'IBE et la PKI suivant divers points

1.3 Bases fondamentales de l'IBE

Cette thèse à pour but d'examiner l'IBE comme technologie évolutive. Pour la bien décrire, nous allons citer tout ce qui la caractérise, à savoir : corps fini, courbes elliptiques, couplages et fonctions de hachage.

1.3.1 Propriétés principales d'un corps fini

Soit p un nombre premier, un corps fini à p éléments F_p est un corps premier de caractéristique p isomorph à Z/pZ (= {0, 1, 2,..., p - 2, p - 1 }).

Définition 1.3.1 : *La caractéristique d'un corps est le plus petit entier n non nul tel que le produit de n'importe quel élément du corps par n soit nul, si ce n n'existe pas alors le corps est de caractéristique 0.*

Définition 1.3.2 : *Un polynôme P(X) de degré n est dit irréductible sur F_p s'il n'admet pas de factorisation en sous polynômes de degré petit que n. Autrement dit, si P(X) n'est divisible par aucun polynôme Q(X) de degré compris entre 1 et (n - 1).*

Proposition 1.3.1 : *Soit p un nombre premier. Soit Q un polynôme irréductible de $F_p[X]$. Alors l'anneau quotient K = $F_p[X]/(Q)$ représente un corps fini de cardinal p^{degQ}. Si on note k par p^{degQ}. Alors :*

$$F_p^k \ = \ \{R(X) \in F_p[X], \ \ \textit{tel que : deg(R)} < k\} \tag{1.7}$$

Trois propriétés d'un corps fini sont vues comme fondamentales et dont on a fait usage dans le chapitre 5, nous les citons dans la proposition 1.3.2.

Proposition 1.3.2 :

1. *Soit K un corps fini de cardinal q = p^d. Soit x ∈ K, alors $x^q = x$.*
2. *Soit L une extension de degré n de K. L'application* $\psi \begin{cases} \text{L} \longrightarrow \text{L} \\ \text{x} \longrightarrow x^q \end{cases}$ *est un automorphism de corps K. De plus, ψ^n = id_L. Ça veut dire que ψ est automorphism de Frobenius sur K.*
3. *Pour q une puissance d'un nombre premier p, le polynôme $X^{q^n} - X \in F_q[X]$ est le produit de tous les polynômes irréductibles de $F_q[X]$ de degré divisant n.*

Arithmétique dans le corps fini

Regrouper des calculs cryptographiques demande d'utiliser des paramètres de grande taille, alors les arithmétiques : multiplication, division, exponentiation ne sont plus faciles. Dans le cas des corps simples, il est conseillé d'utiliser les arithmétiques regroupées dans l'ensuit tableau :

Opération & Algorithme	Complexité	Seuil
Addition	O(n) = A_n	-
Soustraction	O(n)	-
Multiplication M(n) : Scolaire	O(n^2)	0-17
Karatsuba	O($n^{log_2(3)}$)	18-64
Toom Cook 3	O($n^{log_2(5)}$)	65-165
FFT	k $log_2(k)M_w$ + $2klog_2(k)A_p$	256
Division	$M_{m,n}$	-

TABLE 1.2 – Complexités et seuil des algorithmes utilisés pour réduire le taux de calcul dans le cas d'un corps simple.

Pour la méthode FFT qui signifie Fast Fourier Transformation, le k dans la complexité : k $log_2(k)M_w$ + $2klog_2(k)A_p$ représente le degré du polynôme multiplieur, tandis que w est une racine k-ième primitive de l'unité.

Dans le cas d'un corps composé, nous pouvons nous servir du tableau 1.3

Algorithme d'une opération par multi précision	Complexité
Addition	O(n)
Multiplication Classique	2M(n) + M(2n,n)
Multiplication de Montgomery	3M(n)
Multiplication modulaire de Barrett	3M(n)
Multiplication modulaire bipartite	2M(n,n/2) +M(n/2)

TABLE 1.3 – Complexité des algorithmes fondamentaux utilisés pour réduire le taux d'une multiplication modulaire.

Il est alors usuel que la méthode classique est la plus complexe il est déconseillé de l'utiliser Pour la division on utilise souvent l'algorithme d'Euclide Étendu pour les deux cas : corps fini simple et composé. Concernant l'exponentiation les méthodes par Chaîne ou Windows peuvent nous rendre service ; mais dans la plupart des cas on mesure cette opération sur la multiplication scalaire (opération des courbes elliptiques).

1.3.2 Courbes elliptiques

Les courbes elliptiques ont été étudiées en Mathématiques depuis le dix-neuvième siècle. En 1985 et selon deux travaux indépendants, Neal Koblitz et Victor Miller les ont engagées dans la cryptologie (travaux [121] et [135] respectivement). Utiliser les courbes elliptiques dans la cryptographie est privilégié puisqu'elles permettent de travailler avec des clés de longueurs plus courtes par comparaison avec la cryptographie (traditionnelle) basée sur la théorie des nombres, tels que RSA et El-Gamal. Nous matérialisons cela dans le tableau ci-dessous (extrait de [68]) :

Durée de protection	**Sécurité en bit**	**RSA**	**PDL**	**EPDL**
Durée de courte vie	48 - 72	480-1008	480-1008	96-144
Durée moyenne (moins de 2 ans)	80	1248	1248	160
Dix ans de protection	96	1776	1776	192
Vingt ans de protection	112	2432	2432	224
Trente ans de protection	128-192	3248-7936	3248-7936	256-384
Avenir (contre les ordinateurs quantiques)	256	15424	15424	512

TABLE 1.4 – Durée de validité selon le degré de sécurité choisis pour PDL et EPDL.

PDL signifie problème du logarithme discret sur un corps fini.

EDLP signifie problème du logarithme discret sur la courbe elliptique.

Remarques Requises

Sécuriser EDLP, qui est le logarithme discret sur les courbes elliptiques, demande moins de degré de sécurité (on gagne à peu près une réduction de $\frac{1}{8}$) que celui demandé par PDL qui est le logarithme discret sur les corps fini.

Définition 1.3.3 : *Une courbe elliptique définie sur un corps K est l'ensemble des points (x,y) ∈ K (coordonné affine) vérifiant une équation algébrique, plus un point à l'infni. En cas des coordonnées affines la courbe peut être définie par l'équation de Weierstrass :*

$$y^2 + a_1xy + a_3y \;=\; x^3 + a_2x^2 + a_4x + a_6 \tag{1.8}$$

L'équation 1.8 est peut être transformée à une autre sous forme Projective (x=X/Z,y=Y/Z) :

$$Y^2Z + a_1XYZ + a_3YZ^2 \;=\; X^3 + a_2X^2Z + a_6Z^3 \tag{1.9}$$

Ou encore sous forme Jacobienne, en utilisant les coordonnées $(x = X/Z^2, y = Y/Z^3)$:

$$Y^2 + a_1XYZ + a_3YZ^3 \;=\; X^3 + a_2X^2Z^2 + a_4XZ^4 + a_6Z^6 \tag{1.10}$$

Définition 1.3.4 : *Un point P de coordonnées affines (x, y) dans une courbe elliptique E est singulier, si* $\frac{\partial(E)}{\partial(x)}$*=0 et* $\frac{\partial(E)}{\partial(y)}$*=0. La courbe est dite singulière si au moins un de ses points est singulier.*

Une courbe elliptique admet un élément neutre noté universellement par O, qui est de la forme : (0,1,0) sous les coordonnées projectives.

Loi de groupe

Ont dit qu'une courbe elliptique définie sur un corps fini k est munie d'une loi de composition interne additive si :
Soit $P = (x_P, y_P) \in E(k)$ et $Q = (x_Q, y_Q) \in E(k)$, alors :

$$P + O \;=\; P \qquad et \qquad O + P = P \tag{1.11}$$

$$P + (-P) \;=\; O \qquad où \qquad -P = (x_P, -y_P - a_1x_P - a_3) \tag{1.12}$$

Formule explicite :

Soient $P = (x_P, y_P)$ et $Q = (x_Q, y_Q)$. Les coordonnées de P + Q sont définies par :

$$x_{P+Q} = \lambda^2 + a_1\lambda - a_2 - x_P - x_Q, \;\; y_{P+Q} = -(\lambda + a_1)x_{P+Q} - x_P - x_Q - a_3 \tag{1.13}$$

Avec :

$$\lambda = \frac{y_Q - y_P}{x_Q - x_P} \ si \ P \neq Q \qquad et \qquad \lambda = \frac{3x_P^2 + 2a_2x_P + a_4 - a_1y_P}{2y_Q + a_1x_P + a_3} \ sinon \tag{1.14}$$

Interprétation géométrique

Soient P, Q deux points différents de la courbe (différents aussi d'un point à l'infini O). Si on trace la droite (PQ), deux cas peuvent être envisagés :

- La droite coupe la courbe en un troisième point (on démontrera dans le chapitre 5, qu'il y a au plus 3 points d'intersection entre une droite et la courbe). La symétrie de ce troisième point par rapport à l'axe des abscisses est alors P + Q (première courbe de la figure 1.2).
- La droite ne coupe pas la courbe qu'en P et Q. Ce cas n'est possible que si (PQ) est parallèle à l'axe des ordonnées. On définit alors P +Q = O_E (point à l'infini). Si P = Q, on considère la tangente à la courbe en P, P + P est défini comme illustré dans la deuxième figure.

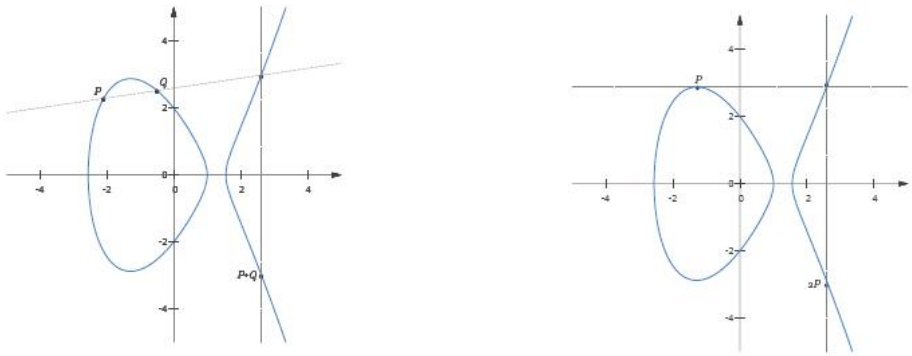


FIGURE 1.2 – Addition et doublement des points de la courbe : $y^2 = x^3 - 5x + 4$

Courbes elliptiques définies sur des corps finis

Dans ce cas les courbes elliptiques E(F_p) sont définies sur un corps fini F_p. La courbe est décrite par l'équation réduite de Weierstrass :

$$y^2 \ = \ x^3 + ax + b(mod \ p) \tag{1.15}$$

a et b appartiennent à F_p et elle est caractérisée par les critères suivants :

$$\Delta = -(4a^3 + 27b^2) \ est \ non \ nul \ et \ le \ j - invariant \ j = -1728 \ \frac{(4a)^3}{\Delta} \tag{1.16}$$

Avec un discriminant non nul on atteint la non singularité d'une courbe elliptique.

Pour un corps k de caractéristique différente de 2 et 3, les coordonnées sont données sous la forme :
Si $x_P \neq x_Q$, P+Q est un point de coordonnées (x_{P+Q}, y_{P+Q}) sachant que :

$$x_{P+Q} \;=\; \lambda^2 - x_P - x_Q \;\; et \;\; y_{P+Q} = \lambda(x_P - x_{P+Q}) - y_P, \;\; avec \;\; \lambda \;=\; \frac{y_Q - y_P}{x_Q - x_P} \quad (1.17)$$

Mais si : $x_P = x_Q$ avec $y_P \neq y_Q$, on peut avoir P+Q=O.
Et si : $y_P = y_Q$, on peut obtenir un point double 2P de coordonnées (x_{2P}, y_{2P}), sachant que :

$$x_{2P} = \lambda^2 - 2x_P \;\; et \;\; y_{2P} = \lambda(x_P - x_{P+Q}) - y_P \;\; avec \;\; \lambda = (3{x_P}^2 + a)(2y_P)^{-1} \quad (1.18)$$

Tenant en compte l'équation d'une courbe elliptique pour un corps de caractéristique différente de 2 et 3, qui est de la forme : $y^2 = x^3 + ax + b$ et ceci après avoir utilisé un changement de variable convenable.

Multiplication scalaire

Pour calculer le produit scalaire kP, on peut utiliser la méthode naïve :

$$\underbrace{P + P + ... + P}_{kfois} \quad (1.19)$$

Mais pour accélérer les calculs, plusieurs algorithmes peuvent être considérés. On cite :

— Méthode Square-and Multiply où kP peut être calculé comme suivant :

$$\text{kP} = \begin{cases} P + 2([\frac{k-1}{2}]P) & \text{si k est impair} \\ 2([\frac{k}{2}]P) & \text{si k est pair} \end{cases}$$

— Méthode Non-adjacent Form (NAF), normalement avec cette méthode on essaye de donner au paramètre k=$\sum_{i=0}^{l} k_i 2^i$ des représentants qui permettent de générer plus de réductions. Avec un NAF normal on choisit $k_i \in$ {-1, 0, 1} et $k_i k_{i+1}$ = 0, alors, il existe au moins un chiffre nul entre deux chiffres non nuls, comme ça, le nombre de $k_i \neq 0$ est minimal ce qui génère moins d'additions.

Comme il faut faire attention aux algorithmes qui accélèrent kP dans les calculs des schémas cryptographiques, il faut aussi virer à la sécurité. Extraire k à partir de kP en connaissant P, demande le déroulement des algorithmes de la cryptanalyse. Il existe des algorithmes dits génériques permettant de trouver k dans un groupe G quelconque, les meilleurs sont : Baby-step Giant-step, Pollard-Rho, Pohlig-Hellman.

1.3.3 Introduction aux couplages

Les couplages sont apparus de par les mathématiciens Weil et Tate au XXième siècle. Leur premier emploi était nuisible puisqu'ils sont utilisés pour attaquer le logarithme discret sur une courbe elliptique [134][78]. La propriété bilinéarité des couplages leur permet de jouer ensuite un rôle constructif [111][22]. L'utilisation des couplages connaît depuis un véritable dévelopement.
Nous donnons ici leurs définitions et leurs principales propriétés, nous aborderons plus de détails au chapitre 5.

Définition 1.3.5 : *Soient G_1, G_2 et G_3 trois groupes abéliens de même ordre r. Les groupes G_1 et G_2 sont additifs, G_3 est un groupe multiplicatif. Un couplage est une application bilinéaire et non dégénérée notée e du produit cartésien de G_1 et G_2 à valeur dans G_3 :*

$$e : G_1 \times G_2 \quad \longrightarrow \quad G_3 \tag{1.20}$$

Un couplage symétrique se réalise lorsque $G_1 = G_2$.

Propriétés des couplages

Un couplage vérifie quatres proporiées dont deux sont essentielles (deux premières)

Non dégénérescence :

$$\forall\, P \in G_1,\ \exists\, Q \in G_2 \ \ tel\ \ que\ \ e(P,Q) \neq 1 \quad et \quad \forall\, Q \in G_2,\ \exists\, P \in G_1 \ \ tel\ \ que\ \ e(P,Q) \neq 1 \tag{1.21}$$

Bilinéarité :

$\forall$ P,P' $\in G_1$, $\forall$ Q,Q' $\in G_2$

$$e(P + P', Q) \ = \ e(P,Q) \times e(P',Q) \tag{1.22}$$

et

$$e(P, Q + Q') \;=\; e(P, Q) \times e(P, Q') \tag{1.23}$$

Identité :

$$\forall P \in G_i,\ i\ \in \{0, 1\} \quad e(P, P) = 1 \tag{1.24}$$

Alternativité :

$$e(P, Q) = e(Q, P)^{-1} \tag{1.25}$$

Algorithme de Miller

Le calcul d'un couplage n'était efficace qu'après l'invention de l'algorithme de Miller, daté de 1986 [136] et développé en 2004 [137], et dont sa présentation est comme suit :

Algorithme 1.3.1 : Miller(P, Q, r) (normal)

Entrée : r = (r_n**...**r_0**)(représentation binaire)**,

P $\in E[r]$(⊂ E(F_q)) et Q $\in G_1$(⊂ $E(F_{q^k})$)

Sortie : $f_{r,P}(Q) \in G_3$ (⊂ $F^*_{q^k}$)

T ← P

$f_1 \leftarrow 1$

Pour i = n - 1 $\grave{a}$ 0 **faire**

1. T ← [2]T

$f_1 \leftarrow {f_1}^2 \times \frac{l_1(Q)}{v_1(Q)}$

l_1 est la tangente à la courbe en T.

V_1 est la verticale à la courbe en [2]T.

2. Si r_i=1 **alors**

$f_1 \leftarrow f_1 \times \frac{l_2(Q)}{v_2(Q)}$

l_2 est la ligne qui passe par TP

V_2 est la verticale au point P + T.

Fin Si

Fin Pour

Retourner f_1

Couplage Symétrique et Asymétrique

Trois types de couplage peuvent être générés : Type 1, 2 et 3.

Type 1 (Couplage Symétrique) :

Il est caractérisé par $G_1 = G_2$

Type 2 (Couplage Asymétrique) :

$G_1 \neq G_2$ mais il existe un homomorphisme calculable et efficace $\phi : G_2 \longrightarrow G_1$

Type 3 (Couplage Asymétrique) :

$G_1 \neq G_2$ et il n'existe aucun homomorphisme calculable et efficace entre G_1 et G_2.

D'après [83], les couplages asymétriques sont les plus convenables à l'utilisation des cryptographes, surtout celui de Type 3. Mais utiliser le type 2 est mieux qu'utiliser le type 1 surtout dans les niveaux de sécurité les plus élevés, nous remarquons cela pour les caractéristiques suivantes montionnées dans [83] : S1(représentation des éléments de G_1), E1 (efficacité des opérations dans le groupe G_1) et F(flexibilité de choisir les systèmes des paramètres).

Quelques exemples des Coupalges

Couplage de Tate

Soit r un entier premier avec la caractéristique p du corps de base F_p.
Soient : K=F_{p^k} un corps qui contient toutes les racines de l'unité (aussi r) ; P un point de $E(K)[r]$; Q un point de E(K). Soient encore : D_P et D_Q deux diviseurs (voir chapitre 5 pour la définition) de degré 0 et de supports disjoints ; f_{D_P} une fonction telle que : $div(f_{D_P}) = rD_P$.
Le couplage de Tate est défini par :

$$t_r : E(F_p)[r] \times E(F_{p^k})/rE(F_{p^k}) \;\; \rightarrow \;\; {F_{p^k}}^*/({F_{p^k}}^*)^r \tag{1.26}$$

$$(P, Q) \;\; \rightarrow \;\; t_r(P, Q) = f_{D_P}(D_Q) \;\; mod\,({F_{p^k}}^*)^r \tag{1.27}$$

Si on prend $D_P = [P] - [O]$, $D_Q = [Q + T] - [T]$ (T est selon notre choix). Alors, D_P et D_Q ont des supports disjoints, donc :

$$t_r(P, Q) \;\; = \;\; \frac{f_P(Q + T)}{f_P(T)} \;\; mod\,({F_{p^k}}^*)^r \tag{1.28}$$

Pour avoir des valeurs exactes, la formule de ce couplage peut s'écrire comme suit :

$$t_r(P, Q) = (f_{D_P}(D_Q))^{(q^k-1)/r} \tag{1.29}$$

Couplage de Ate

Soit E une courbe définie sur F_p qui admet π_p comme application de Frobenius. Le couplage de Ate est défini comme suit :

$$At_r : E(F_{p^k}) \cap Ker(\pi_p - [p]) \times E(F_p)[r] \cap Ker(\pi_p - [1]) \rightarrow {F_{p^k}}^*/({F_{p^k}}^*)^r \tag{1.30}$$

$$(P, Q) \rightarrow At_r(P, Q) = (f_T(D_Q))^{(q^k-1)/r} \tag{1.31}$$

Où T = t - 1, le t est la trace du Frobenius sur E.

Première application constructive du couplage

Les couplages existent depuis le milieu du $XX^{ième}$ siècle, leur premier usage en cryptographie était déconstructif, attaques MOV et FR, en 2000 Joux [111] utilise les couplages pour des fins constructifs en proposant un schéma d'échange de clé tri-partite à la Diffie Hellman bénéficiant de la propriété de bilinéarité des couplages.

Pour générer une clé résultante par le protocole de Diffie-Hellman. L'idée naïve nécessite de faire deux tours (tableau 1.5).

	Alice	Bob	Charlie
Secrets choisis	a	b	c
Premier tour	aP à Bob	bP à Charlie a	cP à Alice
Deuxième tour	[a](cP) à Bob	[b](aP) à Charlie	[c](bP) à Alice
Clé résultante	[a]([c](bP))	[b]([a](cP))	[c]([b](aP))

TABLE 1.5 – Méthode naïve pour générer une clé résultante pour les trois pâtîtes : Alice, Bob et Charlie

En utilisant le couplage, Antoine Joux [111] a converti l'utilisation de ces deux tours à un seul (tableau 1.6).

	Alice	Bob	Charlie
Secrets choisis	a	b	c
Un seul tour	Alice broadcast aP	Bob broadcast bP	Charlie broadcast cP
Clé résultante	$e(bP, cP)^a$	$e(aP, cP)^b$	$e(aP, bP)^c$

TABLE 1.6 – Méthode de Joux pour générer une clé résultante

L'idée de Joux sera assez importante, si on réussit la génération pour un nombre plus que trois. Mais ce problème reste ouvert et personne n'a réussi à le résoudre.

1.3.4 Fonctions de Hachage

Les fonctions de hachage sont premièrement utilisées pour convertir un grand ensemble en un petit ensemble. Elles permettent de créer des empreintes de taille fixe (typiquement, 128, 256 bits) et ceci à partir d'un message arbitraire, ainsi obtenir une fonction à sens unique.

Utilisations

Les fonctions de hachage jouent plusieurs rôles, elles peuvent être utilisées pour associer une chaîne de caractères à un entier particulier, vérifier si une chaîne se trouve ou non dans une table, convertir un message clair à celui qui est arbitraire, signer un message, stockage de mots de passe.

Brève histoire

Les premières fonctions de hachage proposées sont $SHA_{1/2}$ (Secure Hash Algorithm 1 ou 2) et $MDA_{4/5}$ (Message-Digest Algorithm 4 ou 5). Malheureusement ces fonctions tombent respectivement aux attaques [62][120][165]. En 2007 le NIST a lancé un concours pour définir un nouveau standard SHA-3, 14 candidats ont franchi le second tour [177]. Nous renvoyons les intéressés au site [177] pour s'informer sur les plus récents retenus.

Critères d'une fonction de hachage

Pour dire qu'une fonction de hachage est sécurisée, il faut et il suffit qu'elle satisfasse les critères suivants :

Résistance aux attaques en préimage : Étant donné un haché h, trouver M tel que H(M) = h.

Résistance aux attaques en seconde préimage : Étant donné un message M_1, trouver M_2 tel que : $H(M_1) = H(M_2)$.

Résistance aux attaques par collision : Elle consiste $\grave{a}$ trouver deux messages M_1 et M_2 tel que $H(M_1) = H(M_2)$.

1.4 Principe de l'IBE

La gestion infrastructurelle des clés publiques est difficile en PKI. Ceci fut proposé par la solution IBE [161] où chaque personne est identifiée par une chaine unique de bits (en général son identité), à priori n'importe quelle valeur liée à son identité, ça peut être son numéro de sécurité sociale ou son adresse email. Dans ce cas, une autorité génère une clé maîtresse et publie les paramètres du système. On appelle l'autorité dans cette étape KDC

(Key Distribution Center) puisqu'elle prit le rôle de la distribution des clés publiques. Les paramètres publiés sont suffisants à tout utilisateur pour chiffrer un message destiné à une identité choisie (celle du récepteur). A tout moment, après la création par un tiers d'un chiffré à son attention, le récepteur demande à l'autorité la clé privée correspondante à son identité, l'autorité prit dans cette étape le rôle de PKG (Private Key Generateur).

1.4.1 Algorithmes du chiffrement basé sur l'identité IBE

Un schéma de chiffrement basé sur l'identité IBE se définit par quatre algorithmes principaux

<Setup, Extract, Encrypt, Decrypt >

Dans l'algorithme Setup (parfois appelé Params) le KDC regroupe plusieurs calculs, ils concernent les groupes, les fonctions de hachage ainsi que des paramètres supplémentaires. Ces grandeurs seront considérées comme paramètres publics dont le système de chiffrement en besoin, dans Setup le KDC regroupe aussi le calcul de la clé maîtresse qui est une clé publique partagée à tous les utilisateurs. L'algorithme Extract est utilisé par la PKG pour déduire la clé privée d'un utilisateur à partir de son identité. Les algorithmes Encrypt et Decrypt réalisent respectivement le chiffrement et le déchiffrement du message. Nous détaillons encore plus ces algorithmes dans le suivant :

Setup(k) : Prend en entrée un paramètre de sécurité k (selon les regards actuels de la sécurité, par exemple considérons le NIST) et renvoie les paramètres publics mpk et une clé maîtresse secrète msk, normalement cette étape est faite par l'autorité responsable sur les paramètres publics, elle s'appelle KDC ;

Extract(msk,ID) : Prends en entrée la clé maîtresse secrète msk (fixe et utilisée pour toutes les demandes) et l'identité ID de l'utilisateur à qui on compte vise à envoyer le message et renvoie la clé de déchiffrement de l'utilisateur associé à l'identité ID connue en général par usk_{ID}. Une autorité qui s'appelle PKG est la responsable sur cette étape.

Encrypt(mpk,ID,m,s) : Prend en entrée les paramètres publics mpk, l'identité du récepteur ID, le message cible m et un aléa s caractérisant l'émetteur. Cet algorithme renvoie au final un chiffré c du message m associé à l'identité ID.

Decrypt(usk,c) : Prend en entrée la clé secrète usk_{ID} de l'utilisateur où s'adresse le message et un chiffré c. Cet algorithme fait sortir le message m associé au chiffré c sous l'identité ID, et si c ne corresponds pas au chiffré de m sous l'identité utilisé, l'algorithme renvoie un symbole d'erreur $\perp$.

1.5 État de l'art sur les schémas d'IBE

Utiliser l'identité comme clé publique a pour but de simplifier la gestion des clés, en faisant appel à des schémas utilisant des certificats implicites. Cette idée fut inventée par Adi Shamir en 1984 mais sous forme d'un concept. Il a fallut attendre dix sept années c.à.d. jusqu'au 2001 pour voir la première réalisation concrète et sûre des schémas de chiffrement basés sur l'identité.

Brève histoire sur les schémas qui ont échoué entre 1984 et 2001

Pendant la période 1984 à 2001, de nombreuses propositions ont été suggérées [61], [133], [105], [170]et [169] pour répondre au challenge de Shamir, cependant aucun d'entre eux n'était pleinement satisfaisant. Certaines solutions exigent que les utilisateurs ne soient pas en collision, d'autres nécessitent que la PKG passe un long moment pour chaque demande de génération de clé privée et que certaines solutions nécessitent un matériel résistant à la falsification.

En 2001, deux schémas réussis sont proposés indépendamment par Cocks [55] et Boneh et Franklin [22]. Hélas, le schéma de Cocks concerne uniquement les messages très courts puisqu'il chiffre bit par bit. Tandis que celui de Boneh et Franklin peut être considéré comme une vraie réponse à [161] et il a un succès très particulier (il occupe plus de 900 des sites de google [145]). Son secret de réussite se résume à une utilisation de la bilinéarité des couplages, désormais un successif des schémas basés sur cette propriété ont été donnés. Nous rappellerons d'une manière sélective les plus importantes et soulevées par la littérature.

1.5.1 Schémas choisis cités selon les récentes modifications

Nous citons ici six schémas fondateurs qui encadrent l'IBE, ils sont : Boneh-Franklin, Sakai-Kasahara, deux schémas de Boneh et Boyen (BB1 et BB2), schéma de Water et enfin celui de Gentry. Ces schémas sont les grands axes de l'IBE, tous ceux qui sont proposés après sont uniquement leurs dérivés.

Schéma de Boneh et Franklin (version de Galindo)

Dans un papier accepté lors de la conférence CRYPTO 01, Dan Boneh et Matthew Franklin proposèrent 22 la première réalisation restée depuis célèbre implémentation d'un schéma de chiffrement basé sur l'identité (IBE). Le schéma est sous le modèle Random Oracle et il

a connu plusieurs mutations dans le but de le rendre plus efficace, par exemple, Jonathan Katz et Nan Wang [116] lui ont apporté une simple modification afin d'obtenir une réduction dans sa sécurité. Nuttapong Attrapadung et al [12] continuent leur travail en obtenant une réduction encore plus fine. Michel Scott [159] a essayé de prémunir le problème de la projection sur les courbes elliptiques dont le schéma souffre, Gilles Barth et al [14] ont utilisé ce qu'on appelle CertiCrypt pour bien préciser la preuve de Boneh et Franklin. Dans ce contexte nous citons la version de Galindo [86] qui garantit : la réduction dans l'avantage du schéma de Boneh Franklin, utilise les couplages asymétriques de types II et alors fonctionne avec les courbes ordinaires, une preuve de sécurité qui tient en compte ces couplages. Utiliser ce type de couplage est avantageux selon [83], sachant que le schéma de Boneh Franklin est valable seulement pour les couplages symétriques et donc aux courbes supersingulières.

Schéma de Boneh-Franklin (version complète de Galindo [86])

Setup :. Soit (G_1, G_2, G_T, ψ) des groupes bilinéaires.

Choisir un générateur $P_2 \in G_2$ et mettre $P_1 = \psi(P_2)$.

Après sélectionner s$\longleftarrow Z_p$ et calculer $Q_{pub} = sP_2 \in {G_2}^\star \rightarrow P_{pub} = sP_1 \in {G_1}^*$.

Choisir quatre fonctions de hachage : $H_1 : \{0,1\}^* \longleftarrow {G_2}^\star$, $H_2 : G_T \longleftarrow \{0,1\}^n$,

$H_3 : \{0,1\}^n \times \{0,1\}^n \longleftarrow {Z_p}^*$, $H_4 : \{0,1\}^n \longleftarrow \{0,1\}^n$.

L'espace des messages est $\mathbb{M} = \{0,1\}^n$.

L'espace des ciphertext est $\mathbb{C} = {G_1}^* \times \{0,1\}^n \times \{0,1\}^n$.

Extract : Étant donné un ID $\in \{0,1\}^*$, calculé : $Q_{ID} = H_1(ID)$

La clé privée d_{ID} est : $d_{ID} = sQ_{ID} \in {G_2}^*$

Encrypt : Pour chiffrer m $\in 0,1^n$ sous l'identité ID, calculer :

Q_{ID}= $H_1(ID) \in {G_2}^*$, choisir $\sigma \longleftarrow \{0,1\}^n$, calculer r = $H_3(\sigma, m) \in {Z_p}^\star$, alors :

$$C =< rP_1, \sigma \bigoplus H_2(g_{ID}^r), m \bigoplus H_4(\sigma) >, \quad o\grave{u} \;\; g_{ID} = e(P_{pub}, Q_{ID}) \in G_T$$

Decrypt. Soit C = < U, V,W > $\in \mathbb{C}$ un ciphertext sous l'identité ID.

Pour déchiffrer C en utilisant la clé privée $d_{ID} \in {G_2}^\star$, suivre les étapes :

1. Calculer $V \bigoplus H_2(e(U, d_{ID})) = \sigma$.
2. Calculer $W \bigoplus H_4(\sigma)$ = m.
3. Soit r = $H_3(\sigma$, m).

   Vérifier que U = rP, sinon, rejeter le ciphertext.
4. Faire sortir m.

Schéma de Sakai-Kasahara (version de Chen-Cheng)

Toujours sous le modèle Random Oracle et après deux ans que [22], Sakai-Kasahara projettent leurs idées utilisées dans le contexte du traçage de traitres [138] qui est en collaboration avec Mitsunari, sur l'algorithme d'extraction pour un schéma de chiffrement basé sur l'identité. Le schéma obtenu est aussi efficace et il résout le concept de Shamir.
Le schéma de Sakai et Kasahara a connu par la suite plusieurs variantes [154][53], malheureusement leur sécurité a été formellement difficile à prouver. Chen et Cheng [49] sont les premiers qui ont réussis à prouver la sécurité de ce schéma, leur travail [49] présente aussi quelques relations des problèmes de Diffie-Hellman et il a pris en compte le type II des couplages dans la preuve de sécurité traitée. Nous exposons dans ce qui suit la version de Chen-Cheng et nous en tiendrons compte durant toute cette thèse.

Schéma de Sakai-Kasahara (version complète de Chen et Cheng [49])

Setup. Soit (G_1, G_2, G_T, ψ) des groupes bilinéaires.

Choisir un générateur $P_2 \in G_2$ et soit $P_1 = \psi(P_2)$.

Après avoir sélectionner s$\longleftarrow Z_p$, puis, soit $Q_{pub} = sP_2 \in {G_2}^\star \rightarrow P_{pub} = sP_1 \in {G_1}^*$.

Choisir des fonctions de hachage : $H_1 : 0,1^* \longleftarrow {G_2}^\star$,

$H_2 : G_T \longleftarrow \{0,1\}^n$, $H_3 : \{0,1\}^n \times \{0,1\}^n \longleftarrow {Z_p}^*$, $H_4 : \{0,1\}^n \longleftarrow \{0,1\}^n$.

L'espace des messages est $\mathbb{M} = \{0,1\}^n$.

L'espace des ciphertexts est $\mathbb{C} = {G_1}^* \times \{0,1\}^n \times \{0,1\}^n$.

Extract : Étant donné un identifiant $ID_A \in \{0,1\}^n$ d'une entité A, M_{pk} et M_{sk}.

L'algorithme retourne : $d_A = \frac{1}{s+H_1(ID_A)} P_2$.

Encrypt : Étant donné un plaintext m $\in \mathbb{M}$, ID_A et M_{pk}.

Les étapes suivantes doivent être vérifiées :

1. Prendre un arbitraire $\sigma \in \{0,1\}^n$ et calculer r=$H_3(\sigma, m)$
2. Calculer $Q_A = H_1(ID_A)P_1 + P_{pub}$, g^r=$e(P_1, P_2)^r$

   Le ciphertext est :

$$C = (rQ_A, \sigma \oplus H_2(g^r), m \oplus H_4(\sigma)) = (U, V, W)$$

Decrypt : Étant donné un ciphertext C = (U,V,W)$\in \mathbb{C}$, ID_A, d_A et M_{pk}, suivre les étapes :

1. Calculer g'=e(U, d_A) et $\sigma' = V \oplus H_2(g')$
2. Calculer m'=W $\oplus$ $H_4(\sigma')$ et r'= $H_3(\sigma', m')$
3. Si U $\neq r'(H_1(ID_A)P_1 + P_{pub})$ faire sortir $\perp$, sinon retourner m' comme étant le plaintext

---

Schémas BB1 et BB2 (schémas 1 et 2 de Boneh et Boyen [23])

Pendant les conférences EUROCRYPT 04 [23] et CRYPTO 04 [24], Boneh et Boyen ont ajouté trois schémas à l'archive IBE. Malheureusement celui présenté pendant la conférence CRYPTO est non efficace, il n'a obtenu aucune importance par le corps des cryptographes. Les schémas efficaces sont ceux qui sont présentés lors de la seconde conférence, Ils sont connus dans le monde cryptographie par BB1 et BB2 c.à.d respectivement premier et deuxième schéma de Boneh et Boyen. Ces deux schémas sont simples et flexibles ainsi que sémantiquement sûrs, même si leur preuve de sécurité opère sous sélective ID introduit dans [45], ce modèle mal visé comme étant faible. D'aprés son corps le BB2 est une idée variée de Sakai Kasahara mais présentée sous un modèle autre que Random Oracle, le BB1 est une nouvelle approche qui a une flexibilité particulière. On peut le représenter comme IBE ou HIBE, on peut le donner sous Random Oracle, Sélective ID et aussi modèle Standard il faut uniquement jouer sur H_1. Nous donnons ici la version IBE et sous Random Oracle.

Schéma de Boneh-Boyen BB1(version complète) [25]

Setup : Pour générer un système de paramètres BB1, sélectionner :

$\omega, \alpha, \beta, \gamma \in Z_p$, et faire sortir : params = { P, $P_1 = \alpha P, P_2 = \beta P, v_0 = e(P, \hat{P})^{\omega}$ } $\in {G_1}^3 \times G_t$.

Clé maîtresse k = $(\hat{P}, \omega, \alpha, \beta) \in G_2 \times {Z_p}^4$. Soient :

Deux générateurs respectivement des groupes bilinéaires (G_1, G_2) d'ordres premier p : g_1 et g_2.

Un couplage bilinéaire e : $G_1 \times G_2 \longrightarrow G_t$.

Trois fonctions de hachages considérées comme des randoms oracles :

$H_1 : \{0,1\}^* \longleftarrow Z_p$, $H_2 : G_t \longleftarrow \{0,1\}^n$, $H_3 : G_t \times \{0,1\}^n \times G_1 \times G_2 \longleftarrow Z_p$.

L'espace des messages est : $\mathbb{M} = \{0,1\}^n$.

L'espace des ciphertext est : $\mathbb{C} = \{0,1\}^n \times {G_1}^* \times {G_1}^* \times \{0,1\}^n$.

Extract : Sous une identité ID$\in \{0,1\}^l$, pour extraire une clé privée d_{ID} à partir de la clé maîtresse k, sélectionner un random r$\in Z_p$ et faire sortir :

$$d_{ID} = (d_0 = (\omega + (\alpha H_1(ID) + \beta])r)\hat{P}, d_1 = r\hat{P}).$$

Encrypt : Étant donné un plaintext m $\in \mathbb{M}$, ID_A et M_{pk}, suivre les étapes suivantes :

$$C = \begin{cases} C = \text{m} \bigoplus H_2(k = v_0^s), \\ c_0 = \text{sP}, \\ c_1 = H_1(ID)sP_1 + sP_2, \\ \text{t = s + } H_3(k, c, c_0, c_1) \text{ mod p} \end{cases}$$

Où m ∈ {0,1} représente le message, ID ∈ {0, 1} est l'identité du récepteur et

$s \in Z_p$ est l'entier prenne en random.

Decrypt : Étant donné un ciphertext C et une clé privée $d_{ID} = (d_0, d_1)$.

Calculer :

$$k = \frac{e(c_0, d_0)}{e(c_1, d_1)}, s = t - H_3(k, c, c_0, c_1).$$

Si $(k, c_0) \neq$ = (v_0^s, sP), faire sortir $\perp$, sinon, faire sortir m = c $\bigoplus$ $H_2(k)$.

Schéma de Boneh-Boyen BB2 (version CPA) 23

Setup Faire sortir Msk ⟵ (a,b) et Pub ⟵ (P, P_a = aP, P_b = bP, v = $e(P, \hat{P})$),

pour a, b ∈ F_p qui seront choisis en random.

L'espace des messages est : $\mathbb{M} = \{0,1\}^n$.

L'espace des ciphertext est : $\mathbb{C} = \{0,1\}^n \times {G_1}^* \times {G_1}^*$.

Extract : Étant donné (Msk,Id), les clés privées sont :

$$Pvk_{Id} \longleftarrow (r_{Id} = r, \hat{d_{Id}} = \frac{-1}{a + Id + br}\hat{P}) \ \ pour \ \ r \in F_p$$

Où Msk est la clé maîtresse.

Encrypt : Étant donné (Pub, Id, Msg, s), l'encrypt est :

$$Ctx \longleftarrow (c_0 = Msg.v^s, c_1 = sP_a + sIdP, c_2 = sP_b).$$

Decrypt : Étant donné (Pub, Pvk_{Id}, Ctx), le decrypt est :

$$Msg' \longleftarrow c_0.e(c_1 + r_{Id}c_2, \hat{d_{Id}}) \in G_t.$$

Schéma de Waters (Version de Naccach)

Même si sélective ID est classée parmi la catégorie Standard, c'est un modèle faible. Le premier schéma efficace qui présente le pur modèle Standard est celui de Waters en 2005 [173]. Ce schéma est juste une modification de BB1, muni d'une petite variation au niveau Extract. Au lieu de présenter l'identité dans sa forme simple, Waters la compte comme une chaine de caractères $h_i \prod_{i=1} h_i^{ID_i}$. Ceci coûte pour le schéma d'avoir une clé publique de n+4 paramètres. Une version moins complexe a été donnée par David Naccach [140] dans laquelle l'auteur a utilisé des mots au lieu de l'alphabet. Nous décrivons par la suite cette version.

Schéma de Waters (version de Naccache sous forme CPA) [140]

Setup : L'autorité choisit un paramètre secret $\alpha \in Z_p$ arbitrairement et
un générateur g $\in$ G aussi arbitraire.
Elle calcule la valeur g_1 = αg, elle choisit arbitrairemet $g_2 \in$ G.
L'autorité choisit après une valeur arbitraire (random) u' $\in$ G et
un vecteur de longueur n, U=(u_i) qui sera choisit arbitrairement dans G.
Les paramètres publiques sont : $<$ g,g_1,g_2,u',U $>$. La clé maîtresse est αg_2.
Extract (ou Key Generation) : Soient : v = $(v_1, ..., v_n) \in (\{0,1\}^a)^n$ une identité,
r un arbitraire dans Z_p. La clé privée d_v pour une identité v sera :

$$d_v = (\alpha g_2 + r(u' + \sum_{i=1}^{n} u_i), rg)$$

Encrypt : Un message m sera chiffré pour une identité v comme suit :
Choisir une valeur t $\in Z_p$ arbitrairement. Le ciphertext est :

$$C = (e(g_1, g_2)^t m, t.g, t.(u' + \sum_{i=1}^{n} u_i)))$$

Decrypt : Soit C=(c_1, c_2, c_3) le chiffrement valide pour m sous l'identité v.
Alors C peut être decrypté par d_v=(d_1, d_2) comme suit :

$$c_1 \frac{e(d_2, c_3)}{e(d_1, c_2)} = m$$

Schéma de Gentry

Malheureusement la preuve de sécurité donnée par Waters, nécessite d'utiliser un simulateur artificiel dans le cas où certains attributs ne sont pas vérifiés. Parfois il est en doute de faire confiance à cette approche, ainsi, il est difficile d'évaluer le gain de la probabilité en cas d'échec. Bellare et Ristenpart ont présenté [36] une solution où on ne peut pas utiliser ce simulateur artificiel dans la preuve de sécurité de Waters. Un schéma sémantiquement sûr et qui résiste même aux attaques dites à chiffrés choisis a été proposé en 2006 par Gentry [91]. Ce schéma est un peu plus complexe par rapport aux schémas cités dans ce paragraphe. Auparavant, il a beaucoup d'avantage (réduction temporelle plus fine, paramètres fixes, preuve adaptée aux regards posés par Cramer et Shoup [56] en ce qui concerne une preuve standard), plus de détails se trouvent dans l'article de Gentry [91] et dans la thèse de Malika Izabachène [108]. Ce schéma se représente comme suit :

Schéma de Gentry (version complète) [91]

Setup : La PKG sélectionne un générateur arbitraire $<g, h_1, h_2, h_3>$ et un

random $\alpha \in Z_p$. Elle calcule g_1 = αg $\in$ G.

Elle choisit une fonction de hachage H dans la famille one-way hash universel fonction.

Les paramètres publiques sont : <g,g_1,h_1,h_2,h_3,H>. La clé maîtresse est : α.

Extract : Pour générer une clé privée pour une identité ID $\in Z_p$,

la PKG génère un random $r_{ID,i} \in Z_p$ pour i $\in$ {1,2,3}.

Elle fait sortir une clé privée :

$$d_{ID} = \{(r_{ID,i}, h_{ID,i} : i \in \{1,2,3\}, \quad o\grave{u} \quad h_{ID,i} = \frac{1}{\alpha - ID}(h_i + (r_{ID,i}g)).$$

Si ID = α, le PKG échoue.

Encrypt : Pour crypter m $\in G_T$ en utilisant l'identité ID $\in Z_p$,

l'émeteur génère un random s $\in Z_p$ et il envoit le ciphertext :

$$\text{C} = \begin{cases} \text{u=s}g_1 + (-sID)\text{g}, \\ \text{v=}e(g,g)^s, \\ \text{w=m.}e(g,h_1)^{-s}, \\ \text{y=}e(g,h_2)^s e(g,h_3)^{s\beta} \end{cases}$$

Pour C=(u,v,w,y) il calcule β=H(u,v,w)

Decrypt : Pour decrypter le ciphertext C=(u,v,w,y) avec un ID, le récepteur calcule : β=H(u,v,w),

puis il teste si :

$$y = e(u, h_{ID,2}h_{ID,3}^{\beta})v^{r_{ID,2}+r_{ID,3}\beta}.$$

Si le test échoue, le récepteur fait sortir $\perp$. Sinon, il fait sortir m=w.$e(u, h_{ID,1})v^{r_{ID,1}}$

1.5.2 Classification de Xavier Boyen [26]

Xavier Boyen axiomatise les schémas de l'IBE en trois catégories essentielles [26][25], Full Domain, Commutative Blinding et Inversion approche. Ceci est selon la syntaxe de la clé privée, utilise t-elle une fonction de hachage ?, est-elle sous forme inverse ?, ou est ce qu'elle présente une commutative dans sa forme ? Tout en gardant l'impossibilité d'extraire celle-ci. Chaque schéma se range donc sous une classe suivant la forme de sa clé privée. Mais nous apercevrons au chapitre 3 que ce classement n'est pas définitif puisque nous donnerons un schéma qui ne vérifie aucune de ces catégories (section 3.3).

Catégorie "Full Domain Hash"

Dans cette catégorie la clé privée se représente sous le châssis d'une fonction de hachage, autrement dit en Extract $d_{ID} = H(ID)$. De telle manière qu'il est impossible de calculer par exemple un d_{ID_2} après avoir accéder à d_{ID_1}. Le schéma de Boneh Frankin est le seul classé dans ce cadre. Pour garantir l'impossibilité de claculer le d_{ID} par le public, ce schéma requiert la projection sur une courbe elliptique, ce qui lui représente une grave faiblesse.

Catégorie "Inversion approach"

La clé privée prend cette fois la forme Inverse c.à.d. $\frac{1}{s+H(ID)}$ ou $\frac{1}{s+ID}$(utiliser respectivement la fonction de hachage ou non). Le principe dans cette catégorie est d'utiliser le paramètre secret s connu seulement par la PKG (clé master), on l'inverse lorsqu'on ajoute l'identité pour ne pas extraire ce s. Plusieurs schémas sont sous cette catégorie : Sakai Kasahara et toutes ses variantes, BB2, Gentry.

Catégorie "Commutative Blinding "

L'appelation des Commutative-Blinding provient de deux ou plusieurs secrets, on peut créer des Blinding (choses aveugles) qui peuvent commuter entre eux. Les Commutative-Blinding sont beaucoup plus caractérisées par la linéarité, c'est pourquoi on les appelle Full Linearity. L'avantage que présente le commutative Blinding [26] est qu'il est indépendant du domaine haché (première catégorie) et des problèmes mineurs (deuxième catégorie) sous forme exponentiation tel que BDHIP et ABDHP, mais les schémas qui la représentent sont un peu complexes, nous marquons BB1 et Waters comme schémas de commutative Blinding.

1.6 Structures de sécurité d'un IBE

1.6.1 Études de simulations

Normalement un système à clé publique intégré dans un service Network est ouvert aux adversaires (active et même passive). Alors un adversaire actif ne peut pas se limiter à écouter le système mais il peut faire des activités supplémentaires, il peut par exemple inter-changer des messages avec l'autorité, il peut analyser quelques réponses, il peut essayer de casser quelques termes où le message masqué est basé, comme les problèmes de Diffie Hellman....C'est pourquoi on s'est habitué et dans le cadre de la standardisation que la sécurité d'un schéma à clé publique doit être sous l'appris de ce qu'on appelle études des simulations. Ces études sont introduites en [37], elles sont faites en avance pour tester la rigidité de la sécurité du schéma cible.

Un adversaire peut avoir plusieurs **Buts** comme il peut prendre différents **Modèles** pour réussir son attaque contre un schéma. Le premier but est référencié par One Way (OW) Encryption c.à.d. chiffrement à sens unique, le deuxième est connu par indistinguishability (IND). Alors que pour les Modèles nous citons sémantiques CPA et à chiffrements choisis CCA. Le couplement entre but et modèle donne OW-CPA, OW-CCA, IND-CPA, IND-CCA.
Boneh et Franklin ont élargit ces notions pour le chiffrement basé sur l'identité, de telle manière à avoir OW-ID-CPA, OW-ID-CCA, IND-ID-CPA, IND-ID-CCA.
Pendant ces attaques, on considère normalement un simulateur installé par le testeur (celui qui crée le schéma et qui veut juste le tester), il joue le rôle d'un adversaire. On donne à ce dernier tous ce qu'il demande afin de savoir sa puissance concernant l'attaque du schéma en question.
Selon le type du modèle et le but utilisé et suivant le service mis à la disposition de l'adversaire, le degré de la probabilité pour attaquer le schéma se diffère d'une étude à l'autre.
D'après le principe, l'adversaire peut bénéficier des phases 1 et 2. Après la phase 1 il reçoit le challenge, et en poursuivant sa réponse nous devinons sa menace.

Modèle sémantique selon le but IND

Un schéma est dit sémantiquement sûr dans le sens IND, si un adversaire qui a reçu un texte chiffré par une clé publique ne peut rien extraire du message correspondant. Plus particulièrement, le jeu se poursuit entre le challenger et l'adversaire comme suivant :
Supposons que le challenger donne à l'adversaire une clé publique aléatoire. L'adversaire envoie alors deux messages de longueurs égales M_0 et M_1, et reçoit le chiffrement de M_b de la part du challenger où b est choisi aléatoirement dans $\{0, 1\}$. L'adversaire envoie b_0 et gagne le jeu si b = b_0.
Le schéma où on applique l'étude est dit sémantiquement sûr dans le sens IND si l'adversaire peut gagner le jeu dans un temps non-polynômial avec une probabilité supérieure à $\frac{1}{2}$. Nous confirmons dans ce cas que le schéma est IND-CPA sûr.
En premier lieu Boneh et Franklin ont montré [22] la sécurité de leur schéma en se basant sur IND-ID-CPA comme étape première qui aide à comprendre l'étape IND-ID-CCA. À priori pendant IND nous donnons deux messages M_0 et M_1 à l'adversaire et on lui demande de distinguer le bit choisi par le challenger 0 ou 1. Lors d'IND l'adversaire bénéficie de plusieurs dons. Tandis qu'en OW il jouit d'une seule aide.

Nous rappelons ici le IND-ID-CCA comme il est défini par Boneh et Franklin

Notion de IND-ID-CCA

Setup :

Le challenger choisit le système des paramètres et une clé maîtresse, il donne le système à l'adversaire et garde la clé maîtresse (master).

Phase 1 :

L'adversaire adresse des requétes q_1 ....q_m comme suit :

1. **Extraction queries** $\prec ID_i \succ$: Le challenger exécute son algorithme **Extract** pour extraire la clé correspondante $\prec d_i \succ$ de chaque $\prec ID_i \succ$ demandé par l'adversaire et la donne à ce dernier
2. **Decrypt** $\prec ID_i, C_i \succ$ **:** En plus que le challenger exécute l'algorithme **Extract** pour donner la clé correspondante $\prec d_i \succ$ de chaque $\prec ID_i \succ$, il exécute aussi l'algorithme **Decrypt** pour déchiffrer n'importe quelle C_i par sa clé correspondante et il envoie le déchiffré (texte clair) à l'adversaire

Ces clés doivent être choisies adaptatives (c.à.d. chacune dépend de l'autre).

Challenge :

L'ennemi (adversaire) choisit une identité ID et deux messages de même longueur m_{t_1} et m_{t_2} comme test, avec la condition qu'ils ne sont apparus dans aucune des phases 1 et 2.

Le challenger choisit b $\in \{0,1\}$ il envoie C=Encrypt(params, ID, m_{t_b}) à l'adversaire.

Phase 2 :

L'adversaire issue plus de q_{m+1} ....q_n. Les q_i sont :

1. **Extraction queries** $\prec ID_i \succ$ avec $ID_i \neq$ ID de même que la phase 1.
2. **Decrypt** $\prec ID_i, C_i \succ$ **:** Aussi comme la Phase 1 mais avec la condition que $\prec ID_i, C_i \succ \neq \prec ID, C \succ$.

Ces choix doivent être adaptatifs.

Guess :

Finalement l'adversaire répond par $m_{t_{b'}}$ et il gagne si b' = b.

Nous disons que ce test à un avantage adv_t négligeable si

$$adv_t = \mid Pr[b = b'] - \frac{1}{2} \mid < \varepsilon, \quad \forall \varepsilon > 0. \tag{1.32}$$

Le IND-ID-CCA1 utilise la même procédure, sauf que cette fois on n'utilise pas la Phase 2.

Fonction négligeable et algorithme polynômial

Définition 1.6.1 : *Une fonction réelle f est dite négligeable si :*

$$\forall s > 0,\ \exists t_s > 0,\ sachant\ que\ \forall t > t_s,\quad f(t) < \frac{1}{t^s}. \tag{1.33}$$

Définition 1.6.2 : *Un algorithme A est dit en un temps polynomial si A s'exécute dans un temps sous la forme $O(k^s)$, où k représente l'entrée et s est une constante c.à.d $\exists$ une constante positive c et un entier positif t' tel que :*

$$\forall t \geq t',\quad 0 \leq A(t) \leq c.k^s(t). \tag{1.34}$$

Les relations entre les notions citées ci-dessus (celles qui caractérisent les études de simulations) sont souvent données dans la littérature par :

$$\text{IND-ID-CCA2/1} \Rrightarrow \text{IND-ID-CPA} \Rrightarrow \text{IND-ID-OW}$$

Avec le fait que A $\Rrightarrow$ B veut dire si on sait résoudre A on peut aussi résoudre B.

1.6.2 Modèle de sécurité

C'est le modèle utilisé par le schèma cible, il est pris en compte dans sa sécurité. Le genre du modèle utilisé est compté dans le jeu joué entre l'adversaire et le challenger, apprécié comme un algorithme polynômial. Deux modèles jugés comme essentiels dans le domaine sécurité cryptographique : Random Oracle et Standard Modèle.

Modèle Random Oracle

En 1993 Bellare et Rogaway [38] ont introduit le concept du Random Oracle, qui joue le rôle de la fonction de hachage utilisée comme une boite noire (black box) et elle fait sortir un élément random (arbitraire).

• Représentation et définition formelle

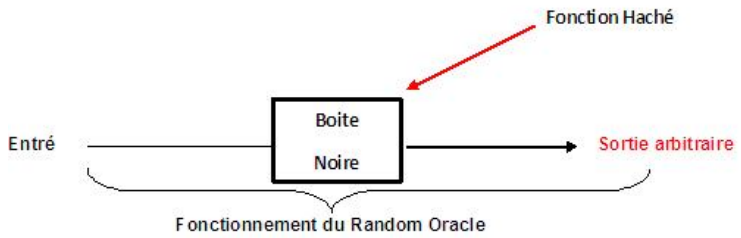


FIGURE 1.3 – Représentation du Random Oracle

Un Random oracle F est l'application

$$\{0,1\}^* \longrightarrow \{0,1\}^\infty$$
$$x \longrightarrow F(x)$$

Le F(x) doit être choisi uniformément et indépendamment pour chaque x. F peut symboliser une fonction de hachage (F=H), dans ce cas l'ensemble de sorties est fixe et il représente un condensé.

Sécurité quelque soit le modèle

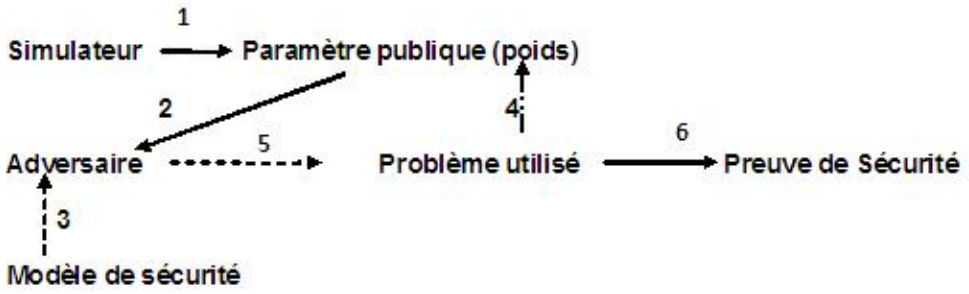


FIGURE 1.4 – Stratégie d'une preuve de sécurité

Le simulateur publie des paramètres publics, l'adversaire prend ces paramètres et suivant le modèle et le poids qu'il a reçu de la part du simulateur, il lance la preuve de sécurité.
Pour prouver la sécurité d'un schéma (quelque soit le modèle), un simulateur met à la disposition d'un adversaire des paramètres publics, considérés comme des dons que l'adversaire peut utiliser pour casser le schéma qu'il a visé à analyser. Ceci peut être réalisé lorsqu'il brise le probléme de Diffie Hellman dans le schéma se base. Pour le faire, l'adversaire demande des requêtes (clés privées dans le cas de CPA et il peut aussi bénéficier de déchiffrements en cas de CCA). Selon le modèle de sécurité utilisé (celui-ci suit la syntaxe de la clé privée), le poids décidé par le simulateur pour le mettre à la disposition de l'adversaire et suivant l'intelligence de ce dernier (méthodes mathématiques suivies par l'adversaire et sa capacité), le schéma peut être attaqué. Autrement, l'adversaire réussit la relation 6 (dans la figure 1.4) c.à.d la preuve de sécurité est vulnérable à l'attaque.
Pour que l'adversaire (nommé algorithme A) maîtrise le jeu, il fait généralement (stratégie suivie par la plupart des preuves et on peut dire tout, surtout Random Oracle et Sélective ID) une coopération avec un tiers nommé algorithme B afin de l'aider à réaliser quelques tâches. L'algorithme B joue le rôle d'un deuxième simulateur par rapport à A, tout en étant un adversaire par rapport au challenger. Pour satisfaire les besoins de A, B déguste les dons qui

lui sont présentés par le premier simulateur (vrai simulateur càd celui qui est dessiné dans la figure 1.4). Au lieu de contacter le vrai simulateur, A communique avec B, ce dernier est obligé de construire les clés privés et de déchiffrer les messages chiffrés selon les demandes de A (suivant l'utilisation du CPA ou du CCA), en utilisant les paramètres publics, sachant que le premier simulateur ne donne pas à B la clé maîtresse.
Construire des clés privées par B et déchiffré des messages est une étape difficile puisque B ne connaît pas la clé maîtresse, qui diffère d'un schéma à l'autre. Selon la syntaxe de la clé privée du schéma testé et la structure des données fournies par le premier simulateur ainsi que le genre du modèle, B peut réussir son étape.

Genre Random Oracle

Comme le Random Oracle est imprévisible par B, ce dernier construit une liste initialement vide. B se familiarise avec le Random Oracle H en utilisant son intelligence et la syntaxe du schéma cible, il se met petit à petit à accomplir la liste selon les demandes de A. Pour plus de détails nous renvoyons au papier de Boneh et Franklin [22].
Il est de coutume de montrer la preuve de sécurité des schémas sous Random Oracle sous CPA et faire ensuite une transformation de Fujisaki-Okamoto pour la rendre CCA sécurisé. Le paragraphe suivant présente cette transformation.

• **Transformation de Fujisaki-Okamoto version [81]**

Fujisaki-Okamoto a suggéré deux versions [81] et [82], la première convertit chaque IND-CPA vers IND-CCA, alors que la deuxième transforme tout OW-CPA à IND-CCA. Mais nous nous intéresserons à la première version dans le chapitre 3. Cette transformation de Fujisaki-Okamoto consiste à utiliser la technique hybride, elle se constitue de quatre axes fondamentaux, par abrégé nous notons hybride par hy.

Setup, Extract, $Encrypt^{hy}$ et $Decrypt^{hy}$

Setup :

Selon un paramètre de sécurité k, cet algorithme génère l'ensemble P des paramètres publics qui engendrent différents paramètres utilisés dans le corps des cryptosystèmes. Il décrit l'espace : des clés K , des messages basiques ($M^{basique}$), espace de quelques paramètres supplémentaires ($\Re$: qu'ils soient sous forme basique ou hybride) et espace des ciphertext, plus, deux fonctions de hachage $H_1 : \{0,1\} \longrightarrow \Re^{basique}$ et $H_3 : \Re^{hy} \longrightarrow K^{sym}$.

Extract

Il génère la clé secrète sk

Encrypt

Dans l'entrée des messages M, $M^{basique}$ et la clé publique pk, l'algorithme sélectionne des arbitraires t et r dans $\Re$ et elle retourne un ciphertext

$$C = Encrypt^{hy}_{pk}(M,t) = (Encrypt^{basique}{}_{pk}(t,r), Encrypt^{sym}{}_{H_3(t)}(M)),\ où\ r = H_1(M||t)$$

Decrypt

Dans l'entré du ciphertext C = (C_1, C_2) en utilisant la clé privée sk, cet algorithme retourne soit un message M ou $\perp$:

Calculer t = $Decrypt^{basique}_{sk}(C_1)$, après calculer M =$Decrypt^{sym}_{H_3(t)}(C_2)$ et r=H_1(M||t)

$Decrypt^{basique}$ et $Decrypt^{sym}$ constituent $Decrypt^{hy}$

Pour savoir la preuve de sécurité qui permet de passer de IND-CPA à IND-CCA, nous renvoyons à l'article original [81].

• **Réduction et révision de Galindo [86]**

Dans un papier publié lors du proceedings ICALP [86], David Galindo a évalué deux choses :

— Trouver une erreur dans la preuve de Boneh Franklin [22]

— Rendre une réduction dans celui ci.

En ce qui concerne l'erreur signalée, il a remarqué que pour générer la liste du Random Oracle H_{liste} dans la preuve de Boneh et Franklin (premier tours-Résultat 1), l'adversaire B a multiplié U_i (premier argument de C_i demandé par A) par un entier alétoire b_i. De telle manière que C'_i=$< b_iU_i, V_i, W_i >$ joue le même rôle que C_i=$< U_i, V_i, W_i >$, puisque $e(b_iU_i, d) = e(b_iU_i, sQ) = e(U_i, sb_iQ) = e(U_i, sQ_i) = e(U_i, d_i)$ c.à.d après avoir utilisé les clés privées d et d_i qui concernent C_i et C'_i respectivement. Ceci n'est pas vrai [86] comme l'ajout de ce b_i dans b_ir_i ($b_iU_i = b_ir_iP$) doit passer par le contrôle d'un autre Random Oracle représenté par la fonction de hachage H_3(σ,M) (voir le schéma de Boneh et Franklin section 1.5.1) et donc il se peut que ce H_3 ne s'adapte pas avec b_ir_i (H_3(σ,M) $\neq b_ir_i$ avec une probabilité 1 - $\frac{1}{p}$) où figure le random b_i, Galindo a remarqué après, qu'avec un $H_1(ID_i)$=Q on ne peut pas avoir ce problème.

Donc, il faut faire attention aux preuves sous Random Oracle puisqu'on utilise toujours le H_3 généré lors de l'utilisation de la transformation de Fujisaki-Okamoto, sachant que la preuve sous ce modèle passe usuellement par ce qui suit :

Pour montrer qu'un schéma est IND-ID-CCA sécurisé, nous aurons besoin de deux schémas BasicPub et BasicPub^{hy}, le premier est un schéma hors IBE, alors que le deuxième est obtenu en utilisant la transformation de Fujisaki Okamoto. La preuve est une conséquence de trois résultats : 1, 2 et 3 comme shématisé au dessous.

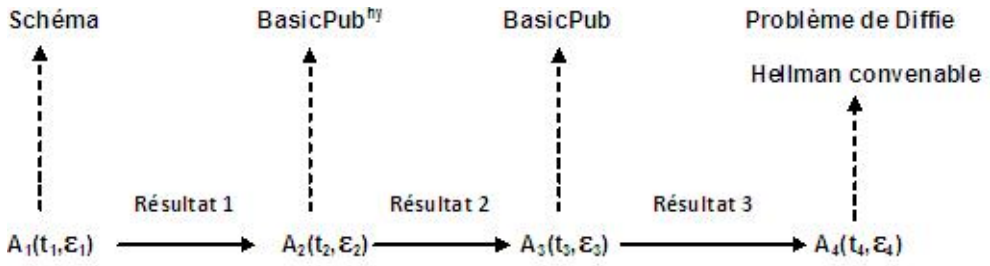


FIGURE 1.5 – Chemin d'une preuve de sécurité d'un schéma d'IBE dans le cas du modèle Random Oracle.

En révisant la faiblesse dans la preuve de Boneh-Franklin, David Galindo a généré une réduction de $\frac{1}{q_H^2}$, sa remarque permet de passer dans le résultat 1 de $\frac{\varepsilon}{(1+q_E+q_D)}$ à $\frac{\varepsilon}{q_H^3}$ (en considérant que $q_E = q_D = q_H$) ce qui est alors plus avantageux. En effet, plus l'avantage du schéma est réduit, moins le schéma est complexe , d'après la simple raison : un avantage grand donne plus de poids pour casser le système ; pour résister il faut augmenter son niveau de sécurité. Par contre, avec un petit avantage on peut utiliser un niveau de sécurité aussi petit que possible en tenant compte par exemple les regards du NIST.

Modèle Standard

Le modèle standard est simplement caractérisé par : aucune utilisation du modèle aléatoire Random Oracle, preuves de sécurité déterministes, généralement les problèmes de Diffie-Hellman utilisés dans les preuves de sécurité sous ce modèle sont décisionnels.

• **Stratégie d'une preuve de sécurité dans le cas du modèle pure**

Généralement, pour montrer la sécurité d'un schéma sémantique on suit la technique des jeux connus aussi par preuve hybride KEM/DEM (Key Encapsulation Mechanisms/DEcapsulasation Mechanisms), Cramer et Shoup sont les premiers qui ont donné cette procédure [56]. La sécurité d'un schéma peut être alors évaluée à travers un jeu d'attaque entre l'attaquant et le challenger. La technique des jeux successifs consiste à tester petit à petit la capacité d'un attaquant suivant des jeux construits, le changement entre eux est suivant trois transitions essentielles.

Transition basée sur l'indistinguabilité : On teste la capacité d'un attaquant s'il peut distinguer un des deux distributions indistinguables.

Transition selon un événement d'échec : Les jeux sucessifs i et i+1 se réalisent de façon identique sauf si un événement d'échec (nommé Ech) se produit. On a :

$$Pr[jeu_i \bigwedge Ech] = Pr[jeu_{i+1} \bigwedge Ech]$$

les jeux sont identiques alors si l'échec ne se produit pas, puisque :

$$Pr[jeu_i \bigwedge Ech] = Pr[jeu_{i+1} \bigwedge Ech] \Longrightarrow Pr[jeu_i] = Pr[jeu_{i+1}] \leq Pr[Ech]$$

Transition complémentaire : C'est une étape intermédiaire qui complète la méthode, cela peut concerner le calcul de certaines quantités.
Par soucis de ces preuves un peu compliquées, cette thèse ne présente aucun schéma et aucune preuve sous le modèle standard pur.

• Preuve de sécurité dans le cas du sélective ID

Le sélective ID est caractérisé par le fait qu'au lieu de choisir l'identité cible à attaquer dans le challenge, on la choisit cette fois dans le setup. Plus d'informations seront données dans le chapitre 3.
La preuve de sécurité cette fois est diffèrente de ce qu'on a décrit précédemment. Dans les deux modèles, Standard pur et Oracle, pour répondre aux phases 1 et 2, on essaye soit d'éliminer la clé maîtresse qui figure dans le poids utilisé par le challenge (voir la figure 1.4) ou encore trouver une méthode qui fait sortir la réponse demandée et ceci selon le problème de Diffie-Hellman pris dans le planning d'attaque (en fait la méthode flexible par l'utilisation de l'attaquant est la responsable sur le problème de Diffie-Hellman utilisé). Dans le modèle sélective ID on utilise ceci et on doit faire attention à l'identité cible attaquée (identité déclarée dans le Setup) si elle ne figure pas dans le calcul fait pendant les phases 1 et 2 et pendant la phase challenge. On essaye de donner une forme à cette identité de telle manière qu'elle apparaît dans le ciphertext.

• Passage d'un CPA à CCA

Pour passer d'un CPA à un CCA (modèle d'attaque) dans le cas du modèle standard, on peut utiliser la méthode one-time signature [45], ou celle de [27] où on ajoute un MAC.
Un MAC est un outil cryptographique qui permet de vérifier l'authenticité de l'origine des données et leur intégrité en même temps. Un MAC est une famille de fonctions de hachage h_k paramétrée par une clé secrète k.

1.6.3 Problème Bilinéaire de Diffie Hellman

Comme nous avons signalé dans ce qui précède, le problème de Diffie-Helman utilisé joue un rôle important dans la preuve de sécurité d'un schéma. A son tour, suivant le poids donné par le premier simulateur (figure 1.4), le problème de Diffie-Hellman prend plusieurs formes :

- Si g^a, g^b et g^c sont donnés comme poids, le défi posé de résoudre devant l'adversaire est le BDHP (cas de Boneh Franklin).
- Si encore le poids est sous la forme : $g^a, g^{a^2}, ..., g^{a^q}$, l'adveraire est appelé à résoudre q-BDHIP (cas de Sakai Kasahara) ou autres problèmes sous forme Exponentiation.
- Si maintenant $g, g', g^a, g^{a^2}, ..., g^{a^q}$ est le poids, l'adversaire est adjuré de résoudre q-ABDHE (cas de Gentry).
- Autres cas

Afin de discuter les problèmes de Diffie et Hellman utilisés dans les preuves de sécurité des schémas qui nous interessent, nous donnons içi, la définition des problèmes CDHP, BDHP utilisés comme fondamentaux dans la cryptographie et nous articulerons les autres selon le besoin.

Problème BDHP

Soient G_1, G_2 deux groupes d'ordre premier q. Soit ê : $G_1 \times G_2 \longrightarrow G_T$ une application admissible et bilinéaire (couplage) et P un générateur de G_1. Le BDHP est comme suit : Étant donné $<$ P, aP, bP, cP $>$ pour a, b, c $\in Z_q$. Calculer ê(P, $P)^{abc} \in G_2$ est difficile.

Problème CDHP

Utilisant le même P, a et b. Étant donné $<$ P, aP, bP $>$, il est très difficile de calculer abP.

Le CDHP est un élément central dans les problèmes de Diffie-Hellman, sa création revient au protocole de Diffie et Hellman [63] (ce protocole est compté ainsi comme base pour créer les problèmes de Diffie et Hellman), il est visible que la sécurité de ce problème dépend du Logarithme Discret (DL).

Problème EDLP

Le problème du logarithme discret (sur les courbes elliptiques) consiste à trouver b à partir de bP et P(donné à l'attaqueur).

Alors si on arrive à extraire b à partir de bP on peut calculer facilement baP (qui représente CDHP).

Ainsi et comme la figure 1.4 l'exprime, le problème de Diffie-Hellman utilisé dépend du poids donné à un adversaire. En plus, le genre du problème influence directement sur la sécurité de

celui ci, puisqu'on peut trouver des problèmes comptés mineurs et ceux classés comme forts. Dans EUROCRYPT 06 Cheon [52] a présenté une faiblesse dans les problèmes sous forme Exponentiation (q-BDHIP, q-BDHEP et autres). Parfois il est difficile de préciser la rigidité d'un problème, en contrepartie il est facile le classer par rapport aux autres. Par exemple d'après la relation traditionnelle :

$$EDLP \Longrightarrow^{1} CDHP \Longrightarrow^{2} BDHP \tag{1.35}$$

Si on sait résoudre EDLP on peut dénouer aussi CDHP, ainsi si on connait CDHP on peut trouver BDHP.

On a justifié la relation 1 (suivant la littérature), la relation 2 est aussi facile, sa preuve est comme suit :

⋆ **Cas** d'un groupe basé sur les courbes elliptiques

Étant donné aP, bP, cP peut on calculer $e(P,P)^{abc}$?

D'après CDHP, on peut calcuer Q=CDHP(aP,bP)=abP, qui nous permet de calculer facilement e(cP,Q) = $e(P,P)^{abc}$

⋆ **Cas** d'un groupe basé sur les corps finis

Étant donné aP, bP et cP on peut calculer e(P,aP), e(P,bP) et e(P,cP). De $h^a = e(P,aP)$ et $h^b = e(P,bP)$ on peut accéder facilement à $Q_h = h^{ab} = CDHP(h^a, h^b)$ et donc à $h^{abc} = e(P,P)^{abc}$ = CDHP($h^{ab} = Q_h, h^c$).

Les relations inverses de l'équation 1.35 ne sont pas prouvées jusqu'à présent, dans le chapitre 5 nous démontrons l'inverse de 2, mais ça sera sous certaines conditions.

Le problème DLP est alors plus fort que CDHP et BDHP et il faut prendre en considération dans un premier temps le niveau qui lui convient dans la sécurité des schémas cryptographiques.

D'autres relations ont été avisées par plusieurs chercheurs [52], [49], [168] et [23]. Autres relations seront abordées dans le chapitre 3.

1.6.4 Schéma sécurisé

Comme récapitulation de tout ce qu'on a vu, un schéma qui passe par toutes les étapes dessinées dans la figure 1.4 est dit sécurisé, si aucun adversaire probable n'a d'avantage non négligeable à travers un temps polynomial. Sachant que l'avantage se calcul comme suit :

$$Adv_A \;=\; |Pr[b = b'] - \frac{1}{2}| \tag{1.36}$$

Chapitre 2

Cryptosytèmes sans Couplage

Quoiqu'apparu en 1984, l'IBE est considéré comme étant un jeune domaine, il n'a été pris au sérieux qu'en 2001 selon deux travaux indépendants, celui de Cocks et le célèbre schéma de Boneh-Franklin. Depuis lors, cette discipline a attiré l'attention des cryptographes, ce qui a donné naissance à plusieurs propositions. Ces dernières se basent sur une notion mathématique qui s'applique sur la bilinéarité s'appelant couplage. Seule exception, le schéma de Cocks qui est basé sur le calcul du symbole de Jacobi, le RSA sous forme IBE proposé par Dan Boneh, Xuhua Ding et Gene Tsudik qui est fondé sur la formule : e $\times$ d=1 mod $\varphi(n)$.
Le présent chapitre est surtout consacré aux schémas de chiffrement sans couplage.
La première partie du chapitre consiste à apporter une valeur au schéma de Cocks en le rendant anonyme. On commence par rappeler en détail les arguments qui permettent de tester l'anonymat de ce schéma, puis, on examine la proposition de Giuseppe Ateniese et Paolo Gasti en citant la preuve qui la met en défaut, une simple variante dans le schéma de Cocks qui permet d'apporter un anonymat efficace à ce dernier est ensuite proposée, un concept qui peut rendre ce schéma universellement anonyme est enfin cité.
La deuxième partie du chapitre consiste à relier l'IBE avec le schéma classique RSA. D'après la littérature, arriver à ce point porte un ensemble de challenges et il est resté sans solutions depuis qu'il a été présenté par Adi Shamir en 1984, Shamir le juge impossible et il s'est limité à le traduire sur les schémas de signature, c'est dans les années 2002 et 2003 où sont citées les seules propositions dans la littérature pour gérer ce problème et ce, de par les cryptographes Boneh, D., Ding, X. et Tsudik, G ([28] et [64]). Malheureusement, ces auteurs ont basé leur idée sur une méthode plus typique qui est celle des SEM.
Nos travaux qui permettent de lier le RSA avec l'IBE commencent par développer l'idée de Boneh, D., Ding, X. et Tsudik, G de telle manière à obtenir un schéma qui nous permet de signer et crypter en même temps, regrouper selon deux versions 1 et 2, la version 2 permet

aussi de résister au Key Escrow. L'étude de sécurité des deux versions est ensuite donnée en tenant compte de la position de l'adversaire qui'est soit interne soit externe. Par la suite, on présente une méthode qui relie l'exposant e utilisé dans le RSA et l'identité ID. Enfin, on propose dans la version 3 une idée qui est totalement différente de celle des SEM et qui conserve le principe du RSA classique.

2.1 Anonymat efficace pour le schéma de Cocks

Le concept anonymat a été en premier lieu introduit par Bellare et al en ASIACRYPT 2001[39]. Ils ont ajouté aux études des simulations la possibilité qu'un adversaire puisse avoir le but de savoir qu'un message est émis par qui et à qui il est envoyé. Il peut par exemple essayer de savoir quelques choses sur la clé utilisée (le principe de Keyword Search). La propriété anonymat à une importance remarquable. En effet, elle peut nous aider à bloquer dès le début l'activité d'un attaqueur ou encore à distinguer les entités mises en communication. Bellare et al ont étudié la propriété de certains crypto systèmes : El Gamal, Cramer-Shoup, RSA-OAEP. Ils ont noté que les deux premiers sont anonymes contrairement au dernier et ils ont apporté un changement convenable pour le rendre anonyme.
En 2005, Halevi [98] ajoute un autre concept à l'anonymat, il concerne la possibilité de distinguer le vrai message de celui qui est arbitraire. On appelle son principe anonymat universel.
Dans un papier publié à EUROCRYPT 2004 [19], Boneh et ses coauteurs signalent que le schéma de Cocks [55] n'est pas anonyme et ce, suite à un test fait par Steven Galbraith lors d'une communication privée. L'anonymat du schéma de Cocks a été dernièrement traité par Giuseppe Ateniese et Paolo Gastien en 2009[10] qui ont essayé d'apporter des modifications appropriées pour le rendre anonyme. En plus, ils ont visé de réaliser un anonymat dit universel. Leur proposition est absolument inefficace soit du point de vue complexité (schéma de longueur plus grande que celle de Cocks) ou anonymat. Esquiver cette faiblesse est l'un des buts de la présente section, nous allons proposer un anonymat efficace pour Cocks. Aussi, nous généralisons la proposition à un anonymat universel.

2.1.1 Préliminaires

Aspects mathématiques

- **Résidu quadratique et résidu non quadratique**

Soit n $\in$ **N**, alors a $\in \mathbf{Z}_n^*$ est un résidu quadratique mod n s'il existe b $\in \mathbf{Z}_n^*$ tel que :

$$a \equiv b^2 (mod\ n) \tag{2.1}$$

Nous notons généralement un résidu quadratique mod n par QR_n et par QNR_n le résidu non quadratique. Avec :

$$QNR_n = \mathbf{Z}_n^* \setminus QR_n \tag{2.2}$$

• **Symbole de Legendre :**

Soit p un nombre premier et a $\in \mathbf{Z}$, $(\frac{a}{p})$ est un Symbole de Legendre qui est définie par :

$$(\frac{a}{p}) = \begin{cases} 0 & \text{p} \mid \text{a} \\ 1 & \text{a} \in QR_p \\ -1 & \text{a} \in QNR_p \end{cases} \tag{2.3}$$

Il vérifie :

$$(\frac{a}{p})(\frac{b}{p}) = (\frac{ab}{p}) \tag{2.4}$$

Où a et b sont deux entiers.

• **Symbole de Jacobi**

Soit n = $\prod_i p^{e_i}$, a $\in \mathbf{Z}_n^*$, le symbole de Jacobi $(\frac{a}{n})$ vérifie :

$$(\frac{a}{n}) = \prod_i (\frac{a}{p})^{e_i} \tag{2.5}$$

et

$$(\frac{ab}{n}) = (\frac{a}{n})(\frac{b}{n}) \tag{2.6}$$

• **Nombre premier de Blum :**

Un nombre premier p tel que : p $\equiv$ 3 mod 4 est nommé nombre premier de Blum.
Parmi les propriétés importantes d'un nombre de Blum sont :

Théorème 2.1.1 : *Soit p un nombre premier, alors :*
-1 $\in QNR_p \iff$ *p est un nombre premier de Blum.*

Théorème 2.1.2 *: Soit n = p.q, produit de deux nombres premiers de Blum. Alors a* $\in QR_n$ *à exactement quatre racines et seulement une d'elles est dans* QR_n*.*

Nous renvoyons le lecteur à [113] pour les preuves de ces deux théorèmes.

Schéma de Cocks [55] :

Schéma de Cocks

Setup : Soit n = pq publié par l'autorité sans être factorisé (principe de RSA) et de 2048 bits, p et q deux nombres premiers de Blum et H une fonction de hachage.

Extract : Pour un ID, l'autorité génère H : $\{0,1\}^* \longrightarrow Z_n$, elle incrémente sa valeur jusqu'à obtenir un symbole de Jacobi $(\frac{H(ID)=a}{n})$ = +1.

La clé secrète correspondante à ID est r tel que :

$$r^2 = a \ \ mod \ \ n \quad où \quad r^2 = -a \ \ mod \ \ n$$

Encrypt : Pour chiffrer un bit m d'un message, nous l'encodons sous la forme b = $(-1)^m$. Le code : +1 pour m=0, et -1 pour m = 1.

Choisir uniformément et arbitrairement des valeurs indépendantes t,v $\in Z_n^*$ telles que :

$$(\frac{t}{n}) = (\frac{v}{n}) = b \quad et \ \ calculé : \quad (c,d) = ((t + \frac{a}{t}) \ \ mod \ \ n, (v - \frac{a}{v}) \ \ mod \ \ n)$$

Decrypt : Pour déchiffrer (c,d), premièrement s = c si r^2 = a mod n et s = d si r^2 = - a mod n. Alors :

$$(\frac{s+2r}{n}) \quad = \quad b. \quad Il \ \ est \ \ simple \ \ de \ \ trouver : \ \ s + 2r = w(1 + \frac{a}{w})^2, \quad avec \ \ w = t \ \ ou \ \ w = v$$

Définition d'un anonymat

Comme nous l'avons souligné, le terme anonymat à une importance remarquable. Vérifier que les schémas populaires sont anonymes ou les transformer de telle sorte qu'ils le deviennent, ceci est plus importants que de créer des nouveaux crypto systèmes [39].

Définition 2.1.1 : *L'anonymat est le fait qu'aucune personne même s'il s'agit d'un adversaire actif, ne peut distinguer le récepteur d'un message. Plus précisément, si un adversaire à deux clés pk_1, pk_2, qui peut accéder à un message chiffré par une de ces deux clés, ne peut pas distinguer avec laquelle d'elles le message a été chiffré.*

Anonymat universel

Définition 2.1.2 [99] : *Est le fait qu'une personne soit incapable de distinguer l'anonymat d'un message chiffré, même s'il utilise la clé publique d'un récepteur.*

Le test de Galbraith (GT) et le non anonymat du schéma de Cocks

Comme c'est révélé en [19] et [10], le schéma de Cocks est non-anonyme, du fait d'un test connu par **T**est de **G**albraith (on le note en abréviation par **GT**) :

Définition 2.1.3 : *Soit* S_a = { $t + \frac{a}{t}$ *mod n /* $t \in \mathbf{Z}_n^*$ } $\cap$ $\mathbf{Z}_n^*$. *Soient deux clés publiques* a_1 *et* a_2, *alors pour distinguer qu'un chiffrement (c,d) du schéma de Cocks est chiffré par* a_1 *ou* a_2 *(c.à.d (c,d) = soit à (* S_{a_1},d_{a_1} *) ou à (* S_{a_2},d_{a_2} *)). Il suffit tout simplement de calculer GT (a,c,n) =* $(\frac{c^2-4a}{n})$ *pour* a_1,$-a_1$,a_2,$-a_2$ *et savoir s'il donne +1. En effet, si* S_a *est chiffré par* a_1 *par exemple, par un simple calcul on obtient :* $(\frac{c^2-4a_1}{n})$ = $(\frac{(t-(\frac{a_1}{t}))^2}{n})$ = *+1.*

Il faut noter que : dire que c est chiffré par a_1 à une probabilité $\frac{1}{2}$.

En plus, selon [113] nous avons si $c^2 - 4a$ est un carré dans Z_n, alors il a quatre racines mod n (théorème 2.1.2). Nous les notons par $\pm$x,$\pm$y, donc :

$$\begin{cases} \text{t} + \frac{a}{t} = \text{c} \\ \text{t} - \frac{a}{t} = \pm\text{x} \quad \text{et} \quad \text{a} \pm\text{y} \end{cases} \Rightarrow t = \frac{c \pm x}{2} \; et \; \frac{c \pm y}{2}. \tag{2.7}$$

Mais trouver les quatre racines, $\pm$x, $\pm$y mod n est un problème très difficile à résoudre, par conséquent, on ne peut pas accéder à a.

Comme conclusion :

$$GT(a,c,n) = \begin{cases} +1 \Rightarrow \text{c} \in S_a \text{ avec une probabilité de } \frac{1}{2} \\ -1 \Rightarrow \text{c n'est pas dans } S_a \end{cases} \tag{2.8}$$

Remarques 2.1.1 :

- *Si GT = +1, l'adversaire peut essayer d'une autre manière ; mais si GT = -1, l'adversaire peut être sûr que le message n'est pas chiffré par la clé cible.*
- ***Le test de Galbraith est le meilleur test pour étudier l'anonymat du schéma de Cocks.*** *En effet, même si le schéma est n'est ni CPA ni CCA sécurisés pour les raisons que nous déclarerons après. On ne peut extraire aucune information sur la clé a, on peut seulement retirer quelques bits du message m. Par contre, suivant 2.8 on peut au moins savoir quelques choses, spécialement si GT (a,c,n) = -1. Donc pour rendre Cocks anonyme, il faut paralyser ce test.*

2.1.2 Variante anonyme du schéma de Cocks

Par une simple variation dans le schéma de Cocks, nous présentons, ci-après, une version anonyme. Nous proposons [3] d'ajouter au niveau Params (génération des paramètres publics dans le Setup) une clé privée K_{pri} munie des paramètres associés.

Par exemple, sous le cadre RSA, la PKG choisit les paramètres : n'= P'Q' non factoriel (p' et q' sont connus seulement par l'autorité) et un exposant e'. Ces deux paramètres sont publiés une seule fois dans un site personnel lié à la PKG. L'émettrice (Alice) sélectionne un paramètre privé K_{pri}, elle le chiffre par e' et l'envoie à l'autorité. Cette dernière qui a une clé d' correspondante à e' peut extraire facilement K_{pri} et le donne au récepteur (Bob) comme clé privée du déchiffrement.

Le paramètre K_{pri} doit être choisi plus grand dans $Z_{n'}$ (n' est au moins de 1024-bits), plus que 512-bits pour qu'il ne puisse pas être connu.

Construction du schéma

Variante du schéma de Cocks

Setup : Soient : n = pq (n peut être égal à n') un nombre non factoriel (de 2048 bits par exemple), p,q deux nombres premiers de Blum et H est une fonction de hachage.

Extract : Pour un ID, l'autorité génère H : $\{0,1\}^* \longrightarrow Z_n$ et elle incrémente sa valeur jusqu'a ce qu'elle obtient un symbole de jacobi $(\frac{H(ID)=a}{n})$ = +1.

La clé secrète correspondante a ID est r tel que :

$$r^2 = a \ \ mod \ \ n \ \ ou \ \ r^2 = -a \ \ mod \ \ n$$

Encrypt : Pour chiffrer un bit m d'un message, nous l'encodons sous la forme b = $(-1)^m$.

Le code : +1 pour m=0, et -1 pour m = 1.

Choisir uniformément et arbitrairement deux valeurs indépendantes t,v $\in Z_n^*$ tel que :

$$\begin{aligned}(\frac{t}{n}) &= (\frac{v}{n}) = b. \quad Et \ \ calculer : \\ (c_1, d_1) &= ((t - K_{pri} + \frac{a}{t}) \ \ mod \ \ n, (v - \frac{a}{v}) \ \ mod \ \ n) \ \ ou \\ (c_2, d_2) &= ((t + K_{pri} + \frac{a}{t}) \ \ mod \ \ n, (v - \frac{a}{v}) \ \ mod \ \ n)\end{aligned}$$

Decrypt : Pour déchiffrer (c_i, d_i),i $\in \{1,2\}$, premièrement s = c_i si r^2 = a mod n et s = d_i si r^2 = - a mod n. Alors :

$$\begin{aligned}(\frac{s + 2r + K_{pri}}{n}) &= b, si \ \ s = c_1 \ \ et \ \ (\frac{s + 2r - K_{pri}}{n}) = b, \ \ si \ \ s = c_2. \\ (\frac{s + 2r}{n}) &= b, \ \ pour \ \ s = d_i, i \in \{1,2\}\end{aligned}$$

Il est simple de trouver que s + 2r $\pm$ K_{pri} = w(1+$\frac{a}{w}$)^2, si s = c_i, pour i $\in$ {1,2} et s + 2r = w(1+$\frac{a}{w}$)^2, si s = d_i, i $\in$ {1,2}. Avec w = t ou v.

2.1.3 Coup d'oeil sur la sécurité du schéma variant de Cocks

La sécurité du schéma variant de Cocks [3] est liée comme celui de Cocks à :

1. La factorisation de n
2. Le problème de calcul de la racine r à partir d'un résidu quadrature a mod n.

Les deux visent de ne pas extraire r. En effet, pour le premier, r=$a^{\frac{\varphi(n)+4}{8}}$ mod n avec $\varphi(n)$=(p-1)(q-1), pour le deuxième r^2=$\pm$ a mod n (le a est publique). Il faut donc être sûr que ces deux problèmes sont difficiles d'être attaqués.
Le rôle de choisir deux nombres premiers de Blum et non d'autres, est de trouver :

$$(\frac{a}{p}) = (\frac{a}{q}) = +1 \;\; si \;\; a \in QR_n \quad et \quad (\frac{-a}{p}) = (\frac{-a}{q}) = +1 \;\; si \;\; a \in QNR_n \tag{2.9}$$

d'après le théorème 2.1.1. Ceci peut alors faciliter le calcul de Jacobi, en plus, on ne peut pas tomber dans l'ambigüité.

Sécurité selon les études de simulations : Le schéma variant de Cocks est comme celui de Cocks, n'est pas CPA sécurisé d'après ce qui est annoncé dans [132]. En effet : $(\frac{c}{n})$ = $(\frac{t^2+a}{n})(\frac{t}{n})$, puisque $(\frac{t}{n})$ est le plaintext de Cocks, alors on aura des précautions très proches sur son bit. Si par exemple le bit de $(\frac{c}{n})$ est -1 alors on est a peu près sûr que le bit de $(\frac{t^2+a}{n})$ soit 1 ($t^2 + a = t^2 + r^2$ ne peut pas être sous forme d'une identité remarquable), cette remarque est aussi valable sur le schéma variant (où $(\frac{c}{n})$ = $(\frac{t^2 \pm K_{pri} t+a}{n})(\frac{t}{n})$).
Il est usuel que cela ne peut pas nous informer sur la clé a.
Aussi, d'après ce qui est annoncé dans [132], le schéma de Cock's (le même sera dit sur le schéma variant), n'est pas CCA2 sécurisé. En effet, si nous assumons qu'une attaqueur Eve à des plaintexts $(m_1, ..., m_k)$ (m_i pour i $\in$ {1,..k}, sont les bits) qui correspondent aux ciphertexts $(c_1, ..., c_k)$, chiffrés à un récepteur Bob. Elle veut chercher les plaintexts qui correspondent aux ciphertext $(c'_1, ..., c'_k)$, elle peut itérativement chercher le plaintext en envoyant à Bob le ciphertext (c'_1 , c_2 , .., c_k). Alors elle note la réaction de Bob sur la clé utilisée, ceci peut rendre Cock's et le schéma variant non anonymes. Mais, on peut résoudre ce problème, puisque l'objectif d'utiliser une clé est l'une des conditions d'un certificat standard (certificat de H). Alors on peut y ajouter le nom du récepteur.
Pour chaque récepteur on peut calculer la première clé et pour les autres, on incrémente H, doubler la valeur et calculer le résultat obtenu suivant le module n. Cela a pour objet d'obtenir des clés personnelles et différentes l'une de l'autre. Plus précisément, si Alice (émettrice) veut contacter Bob et Eve, elle doit déclarer ces deux récepteurs à l'autorité. Cette dernière calcule $H(ID_{receiverB})$=a_{Bob} et $H(ID_{receiverE})$ = a_{Eve}. Alors, pour calculer a_{Eve}, l'autorité peut seulement : ajouter (à un H initial) un nombre arbitraire à a_{Bob}, multiplier le nombre obtenu

par un autre nombre (arbitraire et peut être grand que n) et après elle calcule son module par un mod n, elle double la valeur obtenue et elle recalcule le résultat suite aussi à un module n. En poursuivant cela, on peut résoudre le problème, puisque Eve ne peut pas obtenir le plaintext que pour le message chiffré par sa clé a_{Eve}.

2.1.4 Anonymat du schéma variant de Cocks

Tout d'abord, le Test de Galbraith n'influence pas sur le schéma variant, puisque K_{pri} est inconnu. En plus, le calcul de $c^2 - 4a$ donne toujours -1 et cela même si le message est chiffré par a, par exemple, pour c_1, nous avons :

$$c_1^2 - 4a = (t - K_{pri} + \frac{a}{t})^2 - 4a = t^2 + K_{pri}^2 - 2tK_{pri} + (\frac{a}{t})^2 - \frac{a}{t}K_{pri} - 2a \qquad (2.10)$$

Cette expression contient K_{pri}, t qui sont privés (seulement connus par Alice, l'autorité et Bob). Ainsi, elle est dans QNR_n et alors elle a un GT = -1.
En guise de conclusion, même si c_1 est chiffré par a on ne peut pas savoir l'effet du test de Galbraith.

Généralisation du Test de Galbraith sur le schéma variant

Pour généraliser le test de Galbraith, on doit vérifier que g(a) où g est une fonction quelconque, n'influence pas sur $c_1^2 \pm g(a)$. Premièrement, nous essayons de trouver une identité remarquable (les deux premières) après avoir soustrait ou ajouté g(a) à c_1^2. Nous vérifions après si le Symbole de Jacobi est égal a +1. Pour le faire, on peut réécrire c_1^2 par deux méthodes :

$$c_1^2 = \begin{cases} (t + \frac{a}{t})^2 + K_{pri}^2 - 2tK_{pri} - \frac{2a}{t}K_{pri} & [2.a] \\ (t - K_{pri})^2 + (\frac{a}{t})^2 + 2a - \frac{2a}{t}K_{pri} & [2.b] \end{cases} \qquad (2.11)$$

De l'équation [2.a] on n'extrait rien si on ajoute ou soustrait g(a), puisqu'on obtient toujours 2tK_{pri} + $\frac{2a}{t}K_{pri}$. Aussi, de [2.b] on ne peut rien extraire, puisqu'on a : $\frac{2a}{t}K_{pri}$, K_{pri} et t qui sont inconnus.
Même test est valide pour c_2.
Au contraire de [10], notre preuve d'anonymat est sûr. Dans le schéma de [10] et pour un message chiffré par a, il apparaît un ensemble de $GT(a, Z_1 - T_i, N)$ = +1 qui ont comme probabilié $\frac{1}{2}$, cette dernière représente la moitié pour un test de a et on peut avoir la deuxième tranche par d'autres méthodes. Par contre, avec notre proposition, même si le message est chiffré par a, on a toujours GT = -1. La méthode [10] retarde un petit peu l'adversaire puisqu'il faut faire plusieurs calculs et tests, il n'influence pas sur l'anonymat.

2.1.5 Porposition d'un anonymat universel

Dans le sens de Helvei [98], Hayashi et al [99] et afin d'apporter un anonymat universel au schéma variant, nous proposons 3 ce qui suit :
Générer dans l'algorithme **Extract** deux fonctions de hachage H_1 et H_2 tel que :

$$H_1(ID_{récepteur}) = a = r^2 \ mod \ n, \ \ où \ \ (\frac{a}{n}) = 1. \tag{2.12}$$

$$H_2(ID_{recepteur}) = g \tag{2.13}$$

où g est une fonction inversible d'inverse g'.
Le produit gg', vérifie gg' = 1 mod (p - λ)(q - β), avec (λ,β) $\in Z_n \times Z_n$ sont arbitrairement choisis par l'autorité qui les change pour chaque entité.
Le H_2 doit être une fonction à sens unique, son unique inverse dépend de (p - λ)(q - β), il est seulement connu par l'autorité.
À titre d'exemple on peut prendre :

Exemple 2.1.1 :

g = {e / x mod n $\longrightarrow x^e$ mod n}, avec $H_2(ID_{recepteur})$ = g et g' = {d / y mod n $\longrightarrow y^d$ mod n}
Tel que : gg' = e.d = 1 mod (p - λ)(q - β), pour un λ = -2k_1+1 et un β = -2k_2+1, où $k_1, k_2 \in Z_n$ sont choisis par l'autorité.
Si $k_1 = k_2$ = 0, on a bien : e.d = 1 mod $\varphi(n)$.
Sinon. Soit t $\in Z_n$, alors on a :

$$t^{e.d} = t.t^{k(p+2k_1-1)(q+2k_2-1)} \tag{2.14}$$

$$= t.t^{k(p+2k_1-1)(q-1)}.t^{k(p+2k_1-1)2k_2} \tag{2.15}$$

$$= t.t^{k(p-1)(q-1)}.t^{2k(k_1(q-1)+(p+2k_1-1)k_2)} \tag{2.16}$$

Pour éviter le calcul de : $t^{2k(k_1(q-1)+(p+2k_1-1)k_2}$, l'autorité fixe k_1 et elle cherche un $k_2 \in$ Z tel que : $k_1(q-1) + (p+2k_1-1)k_2$=0 ; à condition que k_2 soit dans Z, sinon il faut changer k_1.
Par conséquent, on obtient : $t^{e.d}$ mod n = t.$t^{k(p-1)(q-1)}$ mod n = t, comme dans le RSA.
On peut aussi refaire les calculs d'une autre manière, puisque le symbole de Jacobi de $t^{2k(k_1(q-1)+(p+2k_1-1)k_2)}$ est égal à : +1.

Alors, on peut écrire :

$$gg' = \{ed = 1 \ mod \ (p-\lambda)(q-\beta)/x \in Z_n \rightarrow x^{ed} = x.cte_x \in Z_n\} \tag{2.17}$$

où le symbole de Jacobi de cte_x est égal a +1.

Dans les deux cas, il faut s'assurer (l'autorité) que (p - λ)(q - β) contient p_λ,q_β tel que : $p_\lambda.q_\beta$ est non factoriel.

Corps du schéma proposé

Setup : Générer deux paramètres publics : g, a et n qui doivent être publiés à Alice (entité qui veut chiffrer un message).

Extract : Générer deux paramètres privés g' et r (pour le récepteur Bob). À condition que : gg' = 1 mod (p - λ)(q - β), pour un λ et β arbitrairement choisis dans $Z_n \times Z_n$ et a = r^2 mod n.

En **Encrypt**, Alice qui veut chiffrer un bit b choisi uniformément et arbitrairement deux valeurs indépendantes t et v telles que : $(\frac{t}{n})$ = $(\frac{v}{n})$ = b et elle calcule :

$$(c_1, d_1) = (g(t) - K_{priv} + \frac{a}{g(t)}, g(v) - \frac{a}{g(v)}) \ ou \ (c_2, d_2) = (g(t) + K_{priv} + \frac{a}{g(t)}, g(v) - \frac{a}{g(v)}) \tag{2.18}$$

En **Decrypt**, pour déchiffrer le message, Bob qui est le récepteur calcule :

$$(g' \times \frac{s + 2r + K_{priv}}{n}) = b, \ pour \ s = c_1, \ (g' \times \frac{s + 2r - K_{priv}}{n}) = b \ pour \ s = c_2 \tag{2.19}$$

Avec :

$$(g' \times \frac{s + 2r}{n}) = b, \ pour \ s = d_i, \ i \in \{1, 2\} \tag{2.20}$$

On remarque que :

$$(\frac{c}{n}) = (\frac{g(t)^2 \pm g(t)K_{priv} + a}{n})(\frac{g(t)}{n}) \tag{2.21}$$

Le $(\frac{g(t)}{n})$ est un chiffrement arbitraire du bit b, alors on ne peut pas extraire $(\frac{t}{n})$ comme on ne sait pas l'inverse de g qui est un problème difficile. Par conséquent, le nouveau schéma est universellement anonyme.

2.2 Le Chiffrement Basé sur l'Identité (IBE) sous forme de RSA

2.2.1 Problématique

En 1984, le cryptographe Adi Shamir a donné naissance à la cryptographie basée sur l'identité (IBC)[161], Il a exploité son idée sur les schémas de signature en utilisant le RSA. Shamir n'a pas réussi à traduire son idée sur les crypto systèmes, en particulier lier celui de RSA avec l'IBE. Pourtant, ceci souffre de plusieurs problèmes, nous citons :

- ✓ Problème de lier le module n ou l'exposant e avec l'identité.
- ✓ Si le module n n'est pas lié à l'identité, il sera difficile de préciser son récepteur convenable.
- ✓ Si l'exposant e est lié à l'identité, il sera difficile qu'il satisfasse les conditions de RSA (càd e factoriel avec $\phi(n)$).
- ✓ Si le module n est universel et fixe pour tout le monde, il peut être fragile contre la factorisation.

Pour toutes ces raisons, l'intégration du RSA dans l'IBE a échoué depuis 1984, malgré qu'il y a quelques propositions faites dans ce sens. Selon nos connaissances, nous marquons deux propositions : idée de Callas [46] et celle de Dan Boneh, Xuhua Ding et Gene Tsudik [28].
Malheureusement l'idée de Jonn Callas est sous forme d'un concept. Dans [46], l'auteur n'a présenté aucune façon d'obtenir ni l'exposant e ni le module n. Surtout l'exposant e qui est lié dans le cas de la cryptographie traditionnelle PKI au récepteur, publié suivant un certificat. Mais dans l'IBE, on vise toujours d'éliminer cette dernière, le [46] n'apporte aucune solution. Ainsi, l'auteur n'a pas parlé sur la méthode de publication de n, est ce qu'on le fixe pour tous les utilisateurs ou on le modifie pour chaque demande. Callas a seulement évoqué le mélange RSA et ID, il n'a présenté aucune réalisation concrète (implémentation). En plus, le [46] est pauvre en études de sécurité et sa réalisation nécessite un effort de plus. C'est pourquoi nous allons nous concentrer dans ce mémoire sur la deuxième idée qui est celle de Dan Boneh, Xuhua Ding et Gene Tsudik [28].

2.2.2 Idée de Dan Boneh, Xuhua Ding et Gene Tsudik : Schéma et Faiblesses

Au début, l'idée de Dan Boneh, Xuhua Ding et Gene Tsudik consiste à résoudre le problème de révocation des clés [20] dans la PKI en utilisant ce qu'on appelle SEM. Les auteurs ont ensuite appliqué l'idée des SEM pour intégrer le RSA dans l'IBE[28]. Leur idée consiste à utiliser

un module fixe et à diviser la clé privée entre deux entités : PKG et SEM. Tenant en compte que le SEM est une entité de confiance, c'est plutôt une dérivée de la PKG. Dans la littérature un RSA avec SEM est noté par mRSA, et IBE avec mRSA est représenté par IB-mRSA. Nous formalisons l'idée [28] dans le schéma suivant :

IBE avec mediat RSA

Setup : Soit k (pair) un paramètre de sécurité

1. Générer deux nombres premiers arbitraires de k/2-bit : p', q' avec p = 2p'+1, q = 2q'+1 sont aussi premiers.
2. n $\longleftarrow$ pq, e $\in_R Z_{\phi(n)}$, d $\longleftarrow e^{-1}$ mod $\phi(n)$
3. Pour chaque utilisateur (Alice) :
   a. s $\longleftarrow$ k - | KG()|-1 (où KG est une fonction de Hachage sous Random Oracle)
   b. $e_{Alice} \longleftarrow 0^s$ || KG(ID_A)||1
   c. $d_{Alice} \longleftarrow \frac{1}{e_{Alice}}$ mod $\phi(n)$
   d. $d_{Alice,u} \longleftarrow^r Z_n$-{0}
   e. $d_{Alice,sem} \longleftarrow$ (d-$d_{Alice,u}$) mod $\phi(n)$

Encrypt :

1. Retirer n, k et KG par un certificat.
2. s $\longleftarrow$ k-|KG()|-1
3. Chiffrer un message m par (e, n) en utilisant RSA/OAEP (spécifié dans PKCS $\neq$ 1v2.1)

Decrypt :

1. USER : m' $\longleftarrow$ message chiffré
2. USER : envoyer m' a SEM
3. En parallèle :
   3.1 SEM :
   (a) Si USER est révoqué retourner (ERREUR)
   (b) $PD_{sem} \longleftarrow m'^{d_{sem}}$ mod n
   (c) Envoyer PD_{sem} à USER
   3.2 USER :
   (a) $PD_u \longleftarrow m'^{d_u}$ mod n

USER : M ⟵ (PD_{sem}.PD_u) mod n

USER : m ⟵ OAEP Decoding de M

USER : S'il réussi, retourner (m)

Malheureusement, l'idée [28] n'est accompagnée d'aucune preuve de sécurité, d'où l'exploitation d'un autre travail [64] présenté par Xuhua Ding et Gene Tsudik dans CT-RSA. Leur preuve de sécurité a montré que IB-mRSA est CCA2 sécurisé dans le modèle des Random Oracle. Certes, ce travail reflète la sécurité, mais porte beaucoup de faiblesses. En effet, dans le papier [64] (même chose est valable pour [28]) l'exposant proposé ne peut pas être dissocié du random oracle, il est proposé sous forme générale, il se peut qu'il ne satisfasse pas la primalité avec $\phi(n)$. Ainsi, l'authentification demandée rend la communication lourde puisqu'il faut contacter à la fois le SEM et la PKG, besoin pour chaque opération du déchiffrement de ce qu'on appelle Token (Jeton). En plus de tout cela, la preuve de sécurité présentée est incomplète.

Quelques Motivations

Comme il a été souligné, intégré le RSA dans l'IBE est encore un problème ouvert. Dans le présent chapitre, nous prenons les solutions existantes [28][64], nous leur apportons des modifications [4] qui les rendent agissantes. Avec un petit changement dans [28][64], nous obtenons un schéma dont lequel nous pouvons signer et crypter en même temps. Cette proposition sera accompagnée de l'exploitation d'un exposant convenable, ainsi que des preuves de sécurité. Au delà, ce travail ne se limite pas à examiner ce qui est écrit dans la littérature ; mais plutôt, à proposer de nouvelles idées [4].

2.2.3 Simple variation dans l'idée [28][64]

Afin de lier le RSA avec l'IBE, les cryptographes Dan Boneh, Xuhua Ding et Gene Tsudik se basent sur deux idées : un module fixe et la division de la clé privée entre SEM et la PKG. Cependant, pour distribuer la clé privée entre ces deux entités, on a besoin d'un canal sûr, ainsi qu'une authentification conjointe par le SEM et la PKG. Cette dernière est peut être coûteuse, puisqu'elle est dans la forme traditionnelle. Par contre, si nous faisons la variation [4] suivante, le SEM déclare et l'autorité vérifie seulement.

- **Description de la variation**

Nous proposons donc que Bob qui n'a pas de clé privée pour déchiffrer le message ${M^{e_B}}_{Alice}$ mod n envoyé par Alice, choisit une clé secrète d_{u_B} et l'envoie à l'autorité via : $d_{uB}^{e_B}$ mod n.

Il est simple pour l'autorité d'extraire d_{u_B} à partir de $d_{u_B}^{e_B}$ mod n, puisqu'elle est la seule qui connaît p,q (les facteurs de n).

Le but de cet envoi est d'informer l'autorité (PKG) que voilà ma clé privée choisie et voilà mon identité, j'ai besoin d'une clé complète pour communiquer. Alors l'autorité prend en considération cette demande, qu'elle stocke quelque part (dans un registre) le couple (d_{u_B}, ID_B) et elle attend le deuxième contact.

Bob contacte ensuite la PKG (message $(H(m), H(m)^{d_{u_B}})$, avec m est un message arbitraire) mais cette fois par l'intermédiaire du SEM.

Le but de ce contact est de permettre au SEM d'authentifier Bob à la PKG.

Alors le SEM informe la PKG, qu'il existe une personne qui demande d_{SEM} et qu'elle peut vérfier son identité. La PKG consulte son registre pour vérifier la clé choisie par Bob, après elle calcule :

$$d_{SEM} = d_B - d_{u_B} \tag{2.22}$$

$$H(m)^{d_{SEM}} \tag{2.23}$$

$$H(m)^{d_{u_B}} \times H(m)^{d_{SEM}} \tag{2.24}$$

$$H'(m) = (H(m)^{d_{u_B}} \times H(m)^{d_{SEM}})^{e_B} \tag{2.25}$$

Après avoir vérifié l'identification de Bob (H '(m) = H (m))

La PKG envoie d_{SEM} au SEM ; mais ceci doit être à travers un canal sûr, sachant que le SEM est une derivée de la PKG.

Le SEM doit s'authentifier devant Bob (message $(H(m), H(m)^{d_{SEM}})$, pour valider sa demande. Alors Bob qui vérifie en plus l'identification du SEM (on a besoin de cette vérification pour ne pas distribuer le message à n'importe quelle personne), envoie le message transmis par Alice (${M_{Alice}}^{e_B}$ muni de $H(m)^{d_{u_B}}$ càd signature) au SEM pour obtenir ce qu'on appelle Token (aide). Ensuite, le SEM qui a reçu d_{SEM} envoie à Bob $({M_{Alice}}^{e_B})^{d_{SEM}}$ mod n.

Comme dernière étape, Bob calcule :

$$({M_{Alice}}^{e_B})^{d_{SEM}} \times ({M_{Alice}}^{e_B})^{d_{u_B}} mod\ n = ({M_{Alice}}^{e_B})^{d_B} mod\ n = M_{Alice} \tag{2.26}$$

Avant d'exposer une méthode convenable (version 1 et 2) d'un IB-mRSA et où nous tiendrons en compte cette variation, tout d'abord, nous donnons une nouvelle proposition d'un exposant lié à l'identité qui sera proposé sous le modèle Standard.

2.2.4 Nouvelle proposition pour un exposant lié à l'identité

Sous une vue traditionnelle, les paramètres publics caractérisant le RSA sont : le module n, l'exposant e. Pour lier ce crypto système avec l'IBE soit que nous lions le module n avec l'idenitié ou encore l'exposant e. Une simple remarque conduit d'interdire la liaison de n, puisque ceci le change pour chaque identité ce qui pose le problème de trouver un convenable $\phi(n)$. Il sera donc plus adapté de lier l'exposant. Les seules propositions présentées (lié e avec ID) dans la littérature sont citées dans [28] et [64]. Même si elles servent la résolution du problème, elles sont mineurs. En effet, elles sont tracées dans le modèle de Random Oracle qui est un modèle aléatoire où on ne peut pas faire confiance, elles ne garantissent pas la primalité avec $\phi(n)$. Tenant en compte cela, ainsi que les perspectives déclarées dans [64]. Notre proposition [4] consiste à tracer l'exposant e dans le modèle Standard.

Tout d'abord, notre but principal est de choisir un e_{ID} qui vérifie les conditions de RSA càd premier avec la fonction d'Euler ϕ(n)=$(p-1)\times(q-1)$. Nous proposons de construire un e_{ID} **premier**, de mesure approximativement $\frac{3}{4}$ la longueur du module n. Effectivement, ce e_{ID} qui est sûrement grand que (p-1) et (q-1) sera aussi premier avec leurs produit (théorème de Bezout). La méthode de construction est comme suit :

1. Projeter le ID dans le code ASCII. Cette projection diffère d'une identité à une autre, puisque tous les ID sont différents (sinon le terme identité n'a pas de sens).
2. Multiplier le résultat par un nombre impair de longueur 2^k (de préférable 2^k+1), avec k est un paramètre approprié et fixe pour tous les ID, de telle manière que le e_{ID} à une longueur $\frac{3}{4}$.n
3. Tester si le nombre est premier ou pas en utilisant les méthodes Miller Rabin [89,150] et AKS[1].

Exemple explicatif (exemple 2.2.1) :

— Prendre un ID et projeter ses caractères sur le code ASCII.

— Si le module est de 2048-bits, selon nos procédures nous souhaitons chercher un e_{ID} de longueur $\frac{3}{4}$.2048= 1536 bits. Comme le ID peut avoir au plus 35 caractères, prenons en compte les caractères d'une organisation (par exemple : @Université Mohamed V.com) que nous pouvons éliminer (or une organisation est la même pour tous les utilisateurs). Cependant, notre ID n'a donc que 15 caractères, càd de 120-bits. Alors, le k qui sera publié par la PKG à 1410 bits ((1536-120-6) bits, les dernier 6 bits (normalement il devrait être un nombre t) sont réservés à la recherche d'un entier premier.

— Nous convertissons la projection de ID dans ASCII en décimal. Si ce nombre est impair (càd qu'il n'a pas de nombre pair dans son LSB en binaire) nous le prenons ; sinon nous lui ajoutons un 1. Nous multiplions ce nombre par exemple par $2^{1410}+1$ et nous testons si ce nombre est premier ou pas. Mais si nous obtenons un nombre non premier, nous lui ajoutons un 2 jusqu'à ce que nous obtenons un nombre premier.

Par cette méthode nous construisons un nombre **premier** plus grand que (p-1) et (q-1) (p et q doivent avoir une taille de longueur $\frac{n}{2}$ afin de ne pas les attaquer), il doit être premier avec (p-1), (q-1) et alors avec leur produit (p-1)×(q-1) (selon Bezout). Comme e_{ID} construit est premier, il doit être rigide contre les attaques de malléabilité [1]. En plus de cela, tous les e_{ID} construits sont différents l'un de l'autre (tous les ID sont différents + multiplication par un nombre fixe 2^k+nombres premiers sont denses).
Cette proposition est efficace, puisque nous obtenons un e_{ID} en module standard tel que le RSA classique. Ainsi on ne peut pas avoir le problème de malléabilité par comparaison avec [28].

2.2.5 Vers une proposition convenable de IB-mRSA : version 1 et 2

Après avoir évoquer quelques limites de la méthode [28] dans la partie 2.2.2, il faut souligner qu'il existe une autre au niveau indépendance. Il s'avère à travers certains auteurs qu'elle a des effets non souhaitables puisqu'elle permet au PKG de lire les messages des utilisateurs, nous pouvons donc imaginer le degré du danger lorsqu'il s'agit des secrets militaires et des entreprises. Pour remédier à ce problème, nous proposons 4 le suivant (deuxième changement dans [28] en utilisant la première variation), qui nous permettra d'obtenir un schéma où on peut signer et crypter en même temps.

Première Version

Demi-Indépendance

Setup : Choisir le paramètre de sécurité n = $p_B \times q_B$, p_B et q_B sont deux nombres premiers très grands (n peut être de 3072 bits, tenant en compte la sécurité actuelle).

Les paramèteres publics sont : $\prec n, H, Z_n \succ$, avec H est une fonction de hachage (de préférable SHA3). Il est recommandé d'obtenir ces paramètres par un certificat.

Nous calculons e_{ID} en utlisant notre méthode (après avoir obtenu le paramètre fixe k).

1. Attaque de malléabilité est le fait d'être capable de faire des laisons entre les réponses misent à la disposition de l'adversaire

Extract : Supposons qu'Alice qui connaît l'identité de Bob et alors sa clé publique, lui envoie un message. Alors Bob qui n'a pas de clé privée, choisit d_{u_B}, il l'envoie à la PKG. Après il calcule $(H(m_B), H(m_B)^{d_{u_B}})$ (le m_B est un message choisi par Bob), il l'envoi au SEM.

Encrypt : Alice (l'expéditrice) choisit $S_{Alice} \in Z_n$.

Elle calcule $((M_{Alice} + S_{Alice})^{e_B}$ mod n,$S_{Alice}^{e_B}$ mod n) = (u,v).

Decrypt : Après la réception de $(H(m_B), H(m_B)^{d_{SEM}})$ du SEM et après une authentification de cette dernière. Bob envoie (u, v) au SEM.

Le SEM répond par $(u^{d_{SEM}}$ mod n,$v^{d_{SEM}}$ mod n)=(u',v').

Après avoir reçu cette réponse, Bob calcule $v'^{d_{u_B}}$ mod n = S_{Alice} et $u'^{d_{u_B}}$ mod n = $M_{Alice} + S_{Alice}$.

Enfin, il calcule la différence $v'^{d_{u_B}}$ mod n-$u'^{d_{u_B}}$ mod n = $M_{Alice} + S_{Alice}$- S_{Alice}=M_{Alice}

Le S_{Alice} choisi par Alice doit être fixe lors de la communication Alice-Bob. Par S_{Alice}, Bob authentifie Alice chaque fois qu'il communique avec elle. En effet, supposons qu'une adversaire Eve veut savoir le contenu de la communication (Alice-Bob). Elle envoie un message M_{Eve} selon ses précautions à propos de la communication entre Alice et Bob. Mais, puisque Eve ne peut pas accéder à S_{Alice} elle choisit S_{Eve} et elle envoie :

$$((M_{Eve} + S_{Eve})^{e_B} mod\ n, S_{Alice}^{e_B}\ mod\ n) \tag{2.27}$$

Logiquement Bob doit aboutir au résultat :

$$(M_{Eve} + S_{Eve})^{e_B} - S_{Alice}^{e_B} \tag{2.28}$$

est un message incohérent, puisque $S_{Eve} \neq S_{Alice}$.

Cela pour le message (Alice-Bob) ; pour l'inverse (Bob-Alice), nous visons qu'à Alice ne contactera aucun centre de sécurité pour obtenir sa clé privée (demi-indépendance). Pour le faire, nous procédons comme suit :

Le S_{Alice} choisi par Alice doit être sous la forme S_{Alice} = $p_{Alice} \times q_{Alice}$ et non factoriel : produit de deux nombres premiers très grands de Z_n (le S_{Alice} par exemple est de 2048-bits)

Alors, après avoir reçu le message par Bob. Pour répondre à ce message, Bob suit la méthode suivante :

Il choisit premièrement S_{Bob} ($*$) dans Z_n. Il envoie le message suivant le module S_{Alice} au lieu de n càd :

$$((M_{R_{Bob}} + S_{Bob})^{e_A}\ \ mod\ S_{Alice}, S_{Bob}^{e_A}\ mod\ S_{Alice}) \tag{2.29}$$

Le $M_{R_{Bob}}$ est la réponse de Bob.

Pour déchiffrer le message, Alice a déjà calculé d_A tel que :

$$d_A \times e_A = 1 \ mod \ \varphi(S_{Alice}) \tag{2.30}$$

Alors elle déchiffre facilement le message.

Le d_A est connu seulement par Alice puisqu'elle est la seule qui connaît p_{Alice},q_{Alice}.

Nous assemblons tous ceci dans la figure (2.1).

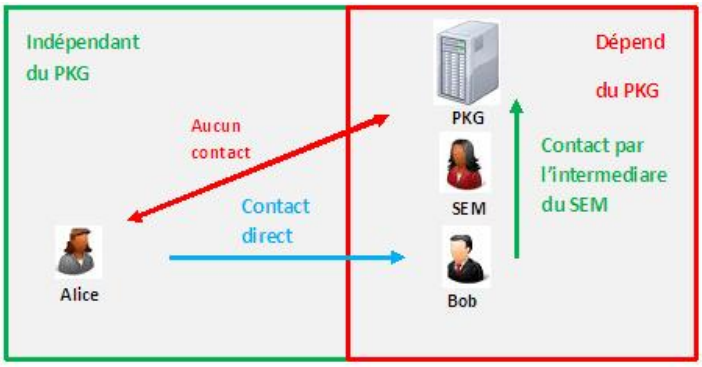


FIGURE 2.1 – Demi-Indépendance (Version 1) :

Dans cette figure, l'émettrice Alice est indépendante de la PKG puisqu'elle génère son propre S_{Alice}. Par contre, Bob ne l'est pas, puisqu'il n'a pas encore généré son paramètre personnel. Cette figure est divisée en deux parties, une qui dépend des SEM et PKG (à gauche tracée en rouge) alors que l'autre ne l'est pas (à droite tracée en vert).

Nous voulons rendre l'indépendance à la partie tracée en rouge càd à celle en gauche.

Indépendance Totale

Si S_{Bob} $(*)$ choisi par Bob dans la section précédente sous la forme : S_{Bob} = $p_{Bob} \times q_{Bob}$ est non factoriel (p_{Bob}, q_{Bob} sont deux nombres premiers connus par Bob seulement).

Alors Bob peut envoyer ses messages sous la forme :

$$((M_{Bob} + S_{Alice})^{e_B} \ mod \ S_{Bob}, \ S_{Alice}^{e_B} \ mod \ S_{Bob}) \tag{2.31}$$

La PKG doit être incapable de lire les messages envoyés par Bob, puisqu'elle ne peut pas factoriser S_{Bob}. De cette manière, Bob peut contacter Alice sans avoir besoin ni du Token (aide) ni de contacter le SEM.

Le problème avec cette technique est que l'adversaire a un champ très vaste de se présenter à la place d'Alice. Alors Bob peut recevoir des messages qui n'ont pas de valeur et il ne peut pas

authentifier ses émetteurs. En plus, Bob peut avoir le problème de révocation de S_{ID_i}. Pour limiter ceci, nous proposons 4 la version suivante.

Seconde Version

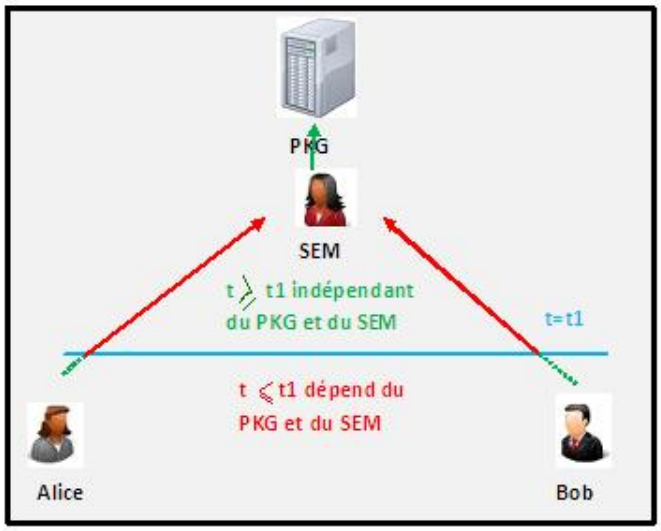


FIGURE 2.2 – Indépendance Totale (Version 2)

Dans la figure 2.2, nous supposons que les deux entités Alice et Bob lors de leurs premiers messages contactent la PKG par l'intermédiaire du SEM. Après un certain moment ils se mettent d'accord pour utiliser leurs paramètres personnels (principe de la cryptographie hybride, mais dans ce cas nous n'envoyons pas une clé symétrique, nous nous limitons à envoyer les paramètres publics :

$$S_{Alice} = p_{Alice} \times q_{Alice} \quad et \quad S_{Bob} = p_{Bob} \times q_{Bob} \tag{2.32}$$

qui doivent être non factoriels pour t $\leq t_1$. Le p_{Alice}, q_{Alice} et p_{Bob},q_{Bob} sont connus seulement par Alice et Bob respectivement). Alors après t $> t_1$ Alice et Bob sont indépendents de la PKG et ils utilisent leurs paramètres personnels.

2.2.6 Étude de sécurité : première et seconde versions

Analyse de sécurité : notions de base

Le RSA est depuis toujours, considéré comme un dispositif probablement sécurisé car il peut être facilement sujet à une extraction des informations à partir d'un message chiffré. En effet, si l'exposant e est premier, il est alors impair, donc sous la forme 2k+1. Par conséquent,

le symbole de Jacobi de y = x^e mod n est égal à celui de x [115], puisque :

$$(\frac{x^e}{n}) = (\frac{x^{2k+1}}{n}) = (\frac{x^{2k+1}}{p})(\frac{x^{2k+1}}{q}) = (\frac{x^{2k}}{p})(\frac{x}{p})(\frac{x^{2k}}{q})(\frac{x}{q}) = (\frac{x}{p})(\frac{x}{q}) = (\frac{x}{n}) \tag{2.33}$$

Mais en réalité ceci n'a pas une grande influence, puisqu'on calcule suivant un module. D'ailleurs, la sécurité de RSA souffre d'autres problèmes. Or, à part la recherche de factoriser le module n ou d'inverser l'exposant e [142], le RSA peut aussi être sensible à une attaque exhaustive par le code ASCII, ce qui exige l'utilisation de ce qu'on appelle padding OAEP[40] (un paiding signifie que nous faisons une concaténation par un 0, 1 ou par un nombre arbitraire). Suite à son rôle dans la sécurité de RSA et à ses amples références [152][34], nous l'utilisons durant tous ce contexte sans formalisation.

Sécurité des deux versions 1 et 2

La sécurité des deux versions 1 et 2 est liée à celle du RSA-OAEP. Celle-ci est originalement CPA sécurisée, pour montrer CCA2, Shoup trouve qu'il ne l'est pas [162]. En 2004, Eiichiro Fujisaki, Tatsuaki Okamoto, David Piontcheval et Jacque Stern ont montré le contraire [76]. Nous n'entrons pas dans les détails (nous renvoyons à [146] pour une large discussion à propos de cela), nous admettons que RSA-OAEP est CCA2 sécurisé ; mais qu'est ce qu'on peut dire de notre crypto système, $\prec (M_{var} + S)^e, S^e) \succ$? (version 1 et 2).
Nous nous limitons à examiner la version 2 (après un certain moment les deux entités Alice et Bob communiquent entre eux indépendamment de la PKG).
Alors les deux entités envoient un message sous la forme $\prec (M + S)^{e_{ID}}$ mod S',$S^{e_{ID}}$ mod S'$\succ$. Nous ajoutons à nos messages, à chaque fois, un nombre constant S (il doit être fixe pour authentifier le transmetteur), pour un nombre suffisant des résultats, notre crypto système, ne peut pas être CPA ni CCA1 et CCA2 sécurisé. Pour remédier à ce problème nous proposons la méthode de masquage suivante :

$$\prec (M + S + h')^{e_{ID}}, (S + h'')^{e_{ID}}, (h' + h'')^{e_{ID}} \succ \tag{2.34}$$

Les fonctions h ', h" sont les masques proposés. Le M + S + h' est égal à cte_i (cte_i est variable pour chaque communication i) et S+h"= cte_i aussi (le h" est variable).
Le déchiffrement de ce message est sous la forme :

$$\prec ((M+S+h')^{e_{ID}})^{d_{ID}}, ((S+h'')^{e_{ID}})^{d_{ID}}, ((h'+h'')^{e_{ID}})^{d_{ID}} \succ = \prec M+S+h', S+h'', h'+h'' \succ \tag{2.35}$$

Alors S + h"-(h '+ h") = S-h', nous soustrairons par 2S (premièrement on doit obtenir S dans un message avant l'indépendance, càd dans un t $< t_1$) nous obtenons : -S-h'
À la fin, nous calculons M+S+h'-S-h'=M.
Après avoir appliqué ce masque, notre crypto sytème munie de OAEP est sous la forme :

$$\prec (((M+S+h')^{e_{ID}}))_{OAEP}, (((S+h'')^{e_{ID}}))_{OAEP}, (((h'+h'')^{e_{ID}})^{d_{ID}})_{OAEP} \succ \quad (2.36)$$

Par exemple $(((M+S+h')^{e_{ID}}))_{OAEP}$ veut dire que nous appliquons le OAEP à : $((M+S+h')^{e_{ID}})$.
Le déchiffrement se fait donc selon la méthode OAEP.
Pour étudier la sécurité de ce crypto système, nous étudions tout d'abord celle de
$\prec (M+S+h')^{e_{ID}}, (S+h'')^{e_{ID}}, (h'+h'')^{e_{ID}} \succ$. Après nous intégrons la sécurité de RSA-OAEP.
Comme nous utilisons la version 2, nous commençons par des messages initiaux :

$$\prec ((m_0+S)^{e_{ID}})_{OAEP}, (S^{e_{ID}})_{OAEP} \succ \quad (2.37)$$

qui doivent être circulés pour t $< t_1$ (dépend de la PKG). Nous sommes intéressés à
$\prec ((m_0+S)^{e_{ID}}, S^{e_{ID}} \succ$ et après nous intégrons le RSA-OAEP.
Nous obtenons alors le système suivant :

$$\begin{cases} \prec (m_0+S)^{e_{ID}} mod(n), S^{e_{ID}} mod(n) \succ \\ \prec (M+S+h')^{e_{ID}} mod(S') = cte_2, (S+h'')^{e_{ID}} mod(S') = cte_2, (h'+h'')^{e_{ID}} mod(S') \succ \end{cases} \quad (2.38)$$

Pour la PKG, c'est un système de 4 équations et 4 inconnus. La PKG ne peut pas déchiffrer ou accéder à :

$$\prec (M+S+h')^{e_{ID}} \ mod\ (S') = cte_2, (S+h'')^{e_{ID}} \ mod\ (S') = cte_2, (h'+h'')^{e_{ID}} \ mod\ (S') \succ \quad (2.39)$$

Mathématiquement il est résoluble.
Pour le public c'est un système de 4 équations et 6 inconnus. Mathématiquement il est irrésoluble.
Peut-on le résoudre ou lui extraire quelques informations si on applique la méthode CPA et CCA (donner des poids aux adversaires) ?
Premièrement nous avons : CCA2 $\longrightarrow$ CCA1 $\longrightarrow$ CPA.
Le terme A $\longrightarrow$ B veut dire que si on sait résoudre A on peut résoudre B. Dans ce cas, le CCA2 est le plus pesant. Alors nous sommes intéressés à montrer la rigidité de l'ancien système contre ce CCA2.

Comme nous sommes dans le cadre de lier le RSA avec l'IBE, nous retenons la définition IND-ID-CCA2 donnée dans[22] et dans le chapitre 1.
La sécurité de la version 2 est liée à :

1. Sécurité de IB^{SEM}-RSA-OAEP
2. Sécurité des multi-utilisateurs
3. Honnêteté de SEM
4. Sécurité contre le key Escrow

Commençons par démontrer cette dernière.

Sécurité contre le Key Escrow

• **Inférence**

Soient (m_{t_1} , m_{t_2}) deux messages de test de même longueur (on peut utiliser des longueurs différentes puisque nous calculons suivant un module). Nous les donnons à la personne qui vérifie l'algorithme de simulation (qui a pris la place d'un adversaire). Différemment de ce qui est en principe (section 1.6 chapitre 1), dans ce qui suit nous donnons à l'adversaire la puissance et la possibilité d'utiliser le **challenge (**m_{t_1}**,**m_{t_2}**)** dans son essai. À titre d'exemple, nous supposons que le challenger choisit m_{t_1}, il le chiffre et l'envoie à l'adversaire virtuel sous la forme :

$$\prec (m_{t_1}+S+h'_{t_1})^{e_{ID}}\ mod\ (S') = cte_2, (S+h''_{t_1})^{e_{ID}}\ mod\ (S') = cte_2, (h'_{t_1}+h''_{t_1})^{e_{ID}}\ mod\ (S') \succ \tag{2.40}$$

Nous visons savoir la réponse de l'adversaire m_{t_1} ou m_{t_2} ?

• **L'adversaire virtuel est le public**

Avant de donner le message de test à l'adversaire il commence son activité ; mais tant que S' est obtenu par la méthode des SEM, il ne peut pas être remplacé. Notre adversaire ne peut pas bénéficier de la première étape dans l'Extract, càd donner des ID et recevoir des d_{ID}. Car il peut extraire facilement $\varphi(s')$. Par conséquent, nous nous concentrons sur la deuxième étape de l'Extract et nous proposons d'utiliser le challenge. Nous utlisons la méthde CPA-CCA2 (dans CCA2 nous ajoutons à la phase 1 et 2 le fait que l'adversaire peut bénéficier de **Encrypt** message. L'adversaire émet donc les demandes (queries) $\prec ID_i, M_i \succ$ de son choix aux challenger, ce dernier les chiffre et l'envoie à l'adversaire). De là, notre adversaire a seulement l'avantage de tester l'expression de $(h' + h'')^{e_{ID}}$ mod (S') qui varie. Mais est ce qu'il peut tirer quelque chose ?. L'adversaire ne peut extraire rien, puisqu'on a une somme où on ne peut pas

séparer le changement de h' de celui de h", en plus le module S' est inconnu. Même avec ceci, on ne peut pas estimer et mesurer la puissance de l'adversaire. Selon le principe, nous lui donnons le challenge (précédent message : le chiffrement de m_{t_1}). Mais tant que l'adversaire à la permission d'avoir plus d'avantages après le challenge et avant sa réponse, il essaie de changer par exemple si on a un adversaire malin.

- ✓ un bit par exemple dans m_{t_1} et deux bits par exemple dans m_{t_2} ;
- ✓ tous les bits sauf un par exemple dans m_{t_1} et tous les bits sauf deux par exemple dans m_{t_2}.

Afin d'apercevoir ceci nous traçons le schéma ci-dessous (figure 2.3)

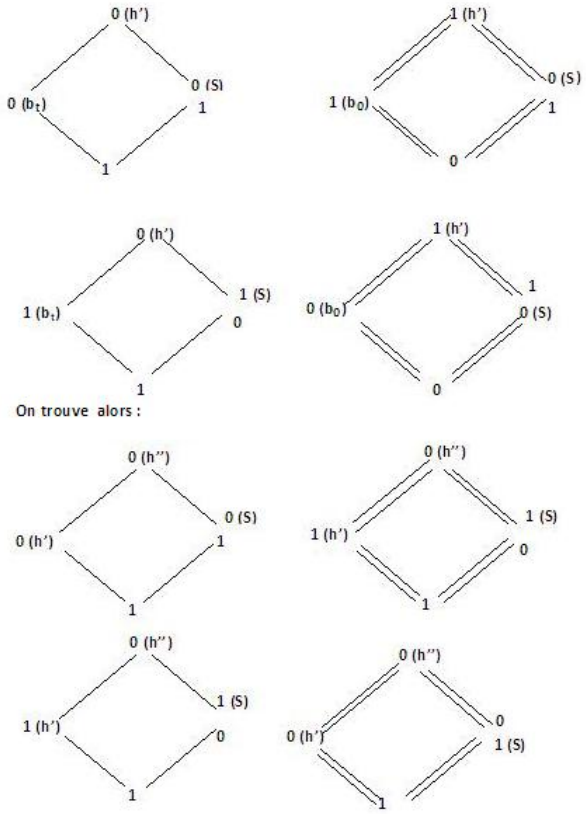


FIGURE 2.3 – Comparaison entre les bits originaux et ceux qu'on varie.

Dans le diagramme de la figure 2.3, les seuls traits sont réservés aux bits qui varient (b_t) et ceux du double ligne sont pour les originels bits (b_0).

Si nous choisissons b_0 = 1 (càd pour l'original bit nous choisissons 1) et si pour h' nous choisissons le bit 0 alors 1(b_0)+0(h') = 1. Mais si nous changeons encore b_0 par b_t nous aurons : 0(b_t)+1(h') = 1 (il faut conserver la même cte=1, comme nous voulons tester seulement h'+h"). Après une addition avec h"=0 nous avons 0(h')+0(h") = 0 pour les originals et 1(h')+0(h") = 1 pour ceux qu'on modifient.

Si nous faisons une comparaison, les résultats s'inversent (le $(h' + h'')_b$ pour les originaux est l'opposé pour ceux du test). Alors, si nous instalons le changement dans LSB ou MSB ou

encore dans $\frac{LSB+MSB}{i}$, on obtient un changement total pour les originaux. Puisque pour LSB le changement influence tous les bits qui restent et il est de même pour $\frac{LSB+MSB}{i}$ (i $<$ au concernant module). Mais si nous installons le changement dans MSB, il peut nous donner un message de petite ou grande longueur que l'original (puisque nous calculons suivant un module, alors la différence des longueurs ne peut pas nous servir). Les résultats seront assez compliqués si nous exposons par e_{ID}, et après avoir calculé suivant un module.
Par conséquence, même si nous avons donné toute cette puissante à notre adversaire (utiliser m_{t_1}, m_{t_2}), nous obtenons un changement total.
En guise de déduction, en plus de S' qui reste inconnu pour notre adversaire public, le $(h' + h'')^{e_{ID}}$ mod(S') est loin d'être attaqué et l'adversaire ne peut pas distinguer si le chiffrement est pour m_{t_1} ou pour m_{t_2} .

• L'adversaire virtuel est la PKG

Dans cette section, notre algorithme de simulation est inspiré à résoudre l'habilité ou plutôt la puissance de la PKG. Nous avons noté précédemment que le système :

$$\begin{cases} \textbf{(1)} & \prec (m_0 + S)^{e_{ID}} \text{ mod (n) }, S^{e_{ID}} \text{ mod (n)} \succ \\ \textbf{(2)} & \prec (M + S + h')^{e_{ID}} \text{ mod (S')=}cte_2, (S + h'')^{e_{ID}} \text{ mod (S')=}cte_2, (h' + h'')^{e_{ID}} \text{ mod(S')} \succ \end{cases} \tag{2.41}$$

La PKG peut le résoudre puisqu'elle a 4 équations et 4 inconnus ; ce qui n'est pas vrai en réalité. Or, même si la PKG peut accéder à **(1)**, donc à (m_0,S) comme nous calculons suivant un mod n ; mais pour **(2)** elle ne peut accèder à aucun de ces paramètres, selon le fait qu'on calcule suivant un mod (S'), le S' ne peut pas être factoriel (la PKG ne peut pas changer S', ce changement peut être visible après être incapable de déchiffrer le message). Alors suivant les arguments cités précédemment, la PKG est incapable d'attaquer le challenge CCA2 puisqu'elle ne peut pas accèder à $(h' + h'')^{e_{ID}}$ mod(S').

Intact sécurité : conditions 1, 2, 3

Pour montrer la sécurité en général (càd conditions 1, 2,3), nous avons besoin des lemmes suivants :

Lemme 2.2.1 : *Le système IB^{SEM}-RSA/OAEP avec un unique utilisateur est polynomial sécurisé tels que le standard RSA/OAEP, càd, $Succ_1 IB^{SEM}(t, q_d) = Succ^R(t', q_d)$ où c est une valeur constante, t' = t + c.*

L'importance de ce lemme est de montrer que pour attaquer le RSA (pour un seul utilisateur) en utilisant SEM est équivalent à l'attaquer dans la forme classique (sans SEM).

Ce lemme est donné dans [28] ; mais nous révisons sa preuve comme nous avons apporté un changement au schéma [28].

Premièrement, il est naturel que $Succ_1^R(t', q_d) < Succ_1 IB^{SEM}(t, q_d)$, en ce qui concerne l'inverse on a :

Il est signalé dans [126] que la preuve [64] n'est pas correcte, comme IB-mRSA/OAEP est probablement insécurisé par une attaque sémantique contre un adversaire interne. Or, la preuve est valable seulement à un RSA sans padding et non pour RSA-OAEP. En effet, dans cette dernière G et H sont des fonctions de Hachage random. La preuve [64] ne tient pas en considération ces fonctions, ce qui demande de penser à une autre manière. Pour remédier à cela, nous recommandons d'utiliser le OAEP dans le modèle standard qui a été prouvé dans [33] qu'il est $NM-CPA sécurisé.

Premièrement nous avons les relations suivantes [87][88] :

$$\$NM - ID - CPA \longrightarrow NM - ID - CPA \longrightarrow IND - ID - CPA \longrightarrow IND - sID - CPA \quad (2.42)$$

($NM-ID-CPA signifie un NM-ID-CPA fort). Donc OAEP est IND-sID-CPA sécurisé, nous allons alors faire notre preuve sous le cadre de cette notion.

Avant de commencer la preuve, nous considérons un adversaire *A* qui travaille sous une sécurité indistinguable.

L'adversaire *A* prend alors la clé publique et il fait sortir deux messages de même longueur m_0 et m_1. Il reçoit un challenge ciphertext C (défi), après avoir prendre sécurisément b $\in_R$ {0,1} en chiffrant m_b.

L'adversaire *A* est chargé de sortir b avec une probabilité non-négligeable et grand que $\frac{1}{2}$.

Pour prouver le lemme 2.2.1, nous considérons un adversaire *B* comme un algorithme chargé d'attaquer l'indistinguabilité dans notre schéma (sous les conditions : clé de fonction de génération, n et 1 càd celui qui contient un unique utilisateur, clé publique e_1 et le secret de paquet d_{u_1}). Dans [64] les auteurs ont permis à *B* de connaître d_{u_1} et ils l'appelle un adversaire malin, dans notre proposition on n'a pas besoin de ceci puisque c'est l'adversaire qui génère d_{u_1}. Comme [64] nous construisons alors *F* comme étant un algorithme contre le standard RSA/OAEP(n̂,ê). L'adversaire *A* enroule *F* comme suit (avec un accès dans le standard-CCA, et il est prouvé dans [88] que : standard-ID-CCA $\leftrightharpoons$ IND-ID-CCA $\longrightarrow$ IND-sID-CPA) :

Experiment F^{RSA}*(n̂,ê) révisé*

Sélectionner deux messages arbitraires (m_0, m_1) et de longueurs égales. Celui qui chiffre sélectionne un bit arbitraire b $\in_R$ {0,1}, il chiffre m_b vers un ciphertext c. Étant donné c, *F* enroule ce qui suit pour obtenir b.

1. Générer un nombre arbitraire d_u et un id ;
2. Initialiser la version 2(n) pour un seul utilisateur qui a n $\longrightarrow$ n̂. Pour un utilisateur $ID_1 \longrightarrow$ id ;
3. $e_1 \longleftarrow$ ê, utilisant notre méthode dans la partie 2.2.4 (càd sans random oracle) ;
4. Initialiser *B*, avec (m_0, m_1, c) est le cible de la version 2(n̂), la clé publique de l'utilisateur est e_1, clé privée : d_u ;
5. Enrouler *B*. Le nombre de clés demandées pour le déchiffrement est majoré par q_d :
   (a) Pour toutes les demandes des clés du déchiffrement de *B*, *F* les donne au simulateur *A*, et il reçoit les réponses (la réponse contient les demandes du déchiffrement + OAEP l'encode/décode) ;
   (b) Pour les demandes c de *B* au SEM : *F* demande le déchiffrement de c. Après avoir obtenu la réponse c^d mod n, *F* calcul c^d/c^{d_u} mod n comme réponse du SEM à *B*.
6. *B* fait sortir b' ;
7. Retourner b' ;

Si b' la sortie de *B* est égale à b, F découvre avec succès b.
L'égalité t'=t + c, a été montré dans [28].

Lemme 2.2.2 : *IB^{SEM}-RSA/OAEP un système de n utilisateurs muni d'un honnête SEM est sémantiquement sécurisé, avec :*

$$Succ_n{}^{IB^{SEM}-RSA}(t_n, q_d, q_e) \leq q_e n Succ_1{}^{IB^{SEM}-RSA}(t_1, q_d)$$

$$où, \quad t_1 = t_n + O(log(q_e n))$$

Nous avons besoin de ce lemme, comme nous utilisons dans IB^{SEM}-RSA un seul module pour des multiples utilisateurs.

Preuve :

Premièrement ce lemme est prouvé dans [64] ; mais il ne satisfait pas notre procédure, puisque la PKG ne peut pas donner à *B* la clé privée. Dans [64] la preuve a été révisée et comme nous n'avons pas utilisé le random oracle dans notre proposition, nous la révisons encore de notre part.

A donne aux multi-utilisateurs la clé publique {n, $e_0, ..., e_n$ }. Nous construisons *B*, pour {n, e = $\prod_{i=0}^{n} e_i$ }

Description

1. $A \longrightarrow B$: (c, e_i),
2. B choisit un d_{u_B} arbitraire, il calcule $c^{d_{u_B}}$
3. $B \longrightarrow$ simulateur du SEM : c'=$c^{d_{u_B}}$.
4. $B \longleftarrow$ simulateur du SEM : $c^{d_{u_B}+d_{SEM_B}} = c^{d_B}$
5. B calcule b = $\frac{e}{e_i}$
6. $A \longleftarrow B$: a = c'^b mod n

Effectivement, la réponse est exactement $c^{\frac{1}{e_i}}$.

Lemme 2.2.3 : *Supposons qu'il existe un IND-sID-CCA adversaire* A *contre notre schéma qui a un avantage* ε *et un temps t(k). Supposons que durant l'attaque* A *fait à peu près* q_d *des demandes de déchiffrement et a peu près* q_e *demandes de chiffrement (permet d'être performé par chaque utilisateur). Alors, il existe un algorithme* A_1 *pour résoudre le Decision RSA Short Encrypted-Prime Problem (D-RSA- SEP) avec un avantage* $\frac{\varepsilon}{2}$ *dans un temps* t_B= t+$q_d(\tau(exp+mod))$
Où $\tau(exp + mod)$ représente le temps maximal pour calculer l'exposentiation et le module.

Le Decision RSA Short Encrypted-Prime Problem étant défini (voir [50] pour les détails) comme suivant :

Définition 2.2.1 de D-RSA- SEP (Decision RSA Short Encrypted-Prime Problem (D-RSA-SEP)) : *Soient* (l_e, l_n, l_k) *des paramètres de sécurité. Étant donné au challenger un triplet* (e', n', z'), *où* n' *est un* l_n*-bit du module RSA,* e' *est un* l_e*-bit premier arbitraire, et* $z' \longleftarrow Z_{n'}{}^*$. *Notre but est de décider si, oui ou non,* z' *est sous la forme* $k^{e'}$ *mod* n', *quelque soit :* k $\in$ *Primes*$(2^{l_{k-1}}, 2^{l_k})$.

Preuve :

Tout d'abord, pour générer un exposant e d'une identité, nous utilisons notre méthode à la place de celle proposée dans [64]. En plus avec notre simple variation dans [28] (qui est valable aussi pour [64]), on peut choisir notre clé privée partielle, alors :
Supposons que *A* à un avantage ε pour attaquer le IB^{SEM}-RSA système. Nous construisons un algorithme A_1 qui utlise *A* pour résoudre le Decision RSA Short Encrypted-Prime Problem (D-RSA- SEP). Le but de l'algorithme A_1 est de faire sortir un 1 s'il est capable de décider si, oui ou non, z_1 est sous la forme k^{e_1} mod n_1, quelque soit k $\in$ Prime $[2^l, 2^{l+1}]$) et 0 si cela n'est pas vrai (en répondant par une valeur arbitraire).

Setup :

Le challenger choisit un n_1 (comme étant le module), une identité ID_1 pour laquelle il calcule e_1 et un l fixe, il publie alors <n_1, ID_1, l>.

Phase 1 :

A demande au plus q_d demandes de déchiffrement. L'algorithme A_1 répond à ces demandes comme suit :

Tout d'abord pour chaque $ID_i \neq ID_1$ qui soit demandés, l'adversaire A_1 utilise notre méthode (partie 2.2.4) pour calculer e_{ID_i}.

Demandes d'extraction d'une clé privée :

A_1 choisit un $d_{partial_{ID_i}}$ il le donne au challenger

A_1 construit une liste de e_{ID_i} et il donne son $d_{partial_{ID_i}}$ au challenger.

Extraction du déchiffrement : Pour chaque ID demandé, A_1 examine sa liste.

Si ID est dans la liste, A_1 demande $c^{d_{SEM_{ID}}}$ du challenger (noté que dans ce cas, le SEM et le challenger sont les mêmes) et il calcule après le déchiffrement des messages.

Sinon, A_1 choisit ensuite $d_{partial_{ID}}$ il le donne au challenger, il reçoit $c^{d_{SEM_{ID}}}$. À la fin, il donne le plaintext à *A*.

Avant de construire le challenge, nous notons que notre système peut s'écrit comme étant :

$$((mh')^e mod\, n, (sh'')^e mod\, n, (h'h'')^e mod\, n) \tag{2.43}$$

À la place de

$$((m + h')^e mod\, n, (s + h'')^e mod\, n, (h' + h'')^e mod\, n) \tag{2.44}$$

Ces deux messages sont les mêmes, la seule différence entre eux est que le premier est un peu long que le deuxième.

Challenge :

L'adversaire *A* décide de terminer la Phase 1, il fait sortir deux plaintexts de longueurs égales m_0, m_1

(A_1 répond avec le ciphertext $C_b = ((m_b)^{e_1}.z_1 \text{ mod } n_1, (sh''_b)^{e_1} \text{ mod } n_1, (k_1 h''_b)^{e_1} \text{ mod } n_1)$

A noter que : A_1 peut calculer n'importe quel arbitraire $(sh''_b)^{e_1}$ mod n_1, après qu'il l'avoir reçu du challenger dans la phase 1.

Alors si $z_1 = {k_1}^{e_1}$ mod n_1, C_b est un valide ciphertext pour M_b

Phase 2 :

On procède comme dans la phase 1, sauf que A_1 n'est pas permi d'envoyer le déchiffrement de ID_{ch}.

Guess :

Enfin, *A* sort son guess b' $\in \{0,1\}$ quand A_1 sort b. Si b = b' alors A_1 sort 1, ce qui signifie que $z_1 = {k_1}^{e_1} \bmod n_1$. Sinon, il sort 0.

1. Si z_1 n'a pas la forme ${k_1}^{e_1} \bmod n_1$, càd, si z_1 est un random dans $Z^*_{n_1}$. Dans ce cas, le ciphertext du challenger n'est pas clair et la simulation est peut être échouée. Alors l'avantage d'un adversaire est necessairement 0, puisque le ciphertext doit être indépendant de m_b. La probabilité d'une réponse échouée est $\frac{1}{2}$.
2. Si $z_1 = k_1^{e_1} \bmod n_1$ quelque soit l_{k1}-bit premier. Alors l'avis d'un adversaire est normal et il doit avoir ε comme avantage.

Comme conclusion, nous avons un avantage $\frac{\varepsilon}{2}$ pour casser le problème *A*-D-RSA-SEP.
Ce lemme est prouvé dans le modèle selective-ID qui est inventé par [45].

Théorème 2.2.1 : *Notre schéma (ou plutôt système) est sécurisé contre IND-sID-CCA adversaire. Supposons qu'il existe un IND-sID-CCA adversaire* A *contre notre schéma et qui a un avantage* Adv *et un temps* t*(k). Alors :*

$$Adv \geq \tfrac{\varepsilon}{2} {Succ_n}^{IB^{SEM}-RSA}(t_n, q_d, q_e)$$

Et

$$t(k) = t_n + q_d(\tau(exp + mod)) = t_1 - O(log(q_e n)) + q_d(\tau(exp + mod))$$

Preuve : *Preuve du Lemme 2.2.1 + Preuve du Lemme 2.2.2 + Preuve du Lemme 2.2.3.* □

2.2.7 Comparaison entre l'authentification utilisée dans les versions 1 et 2 et la signature de Shamir

Le nom de RSA est un abrégé de Rivest Shamir Adelman, c'est à dire que le cryptographe Shamir est l'un de ces inventeurs [151]. Celui-ci qui a aussi donné naissance à la technologie IBE [161], n'a pas réussi à intégrer le RSA dans ce dernier et il l'utilise seulement dans les schémas de signature (chapitre 1, partie 1.2). Cette dernière sert à authentifier les utilisateurs. Ce qu'on a fait durant tout ce contexte est le contraire, on a proposé une méthode [4] pour intégrer le RSA dans l'IBE, qui nous permet de signer et de crypter en même temps. Mais pour authentifier les utilisateurs (signer), on a basé sur l'intégrité des données. Peut-on dire qu'avec cette signature, on peut avoir le même niveau de sécurité que la signature de Shamir ?

Comparaison

Paramètre de la signature de Shamir 161	Paramètre de notre authentification
$\prec s = g.r^{f(t,m)}, t = r^e \succ$	$\prec (m+r)^e, r^e \succ$

Dans notre authentification, on a choisi r à la place de S_{Alice} et S_{Bob} pour simplifier la comparaison.

Pour Signer

— Deuxième terme : Pour les deux, nous avons le même second terme r^e.

— Premier terme : Pour **Shamir**, il contient le paramètre secret g, l'expression de s est liée à : r, m et aux paramètres publics f, t. Pour la **nôtre**, l'expression de $(m+r)^e$ est aussi liée à r et m (le message) plus le public paramètre e.

Alors, notre authentification a à peu près les mêmes paramètres que la signature de Shamir (cette dernière est plus large que la nôtre).

Pour Vérifier

- Nous notons que la vérification de Shamir est liée à i = g^e et t = r^e.
- Alors que notre authentification est basée seulement sur l'intégrité des données et sur l'identification.

Sécurité

- La sécurité de la signature de Shamir est liée à la clé secrète g et r, comme i = g^e et t = r^e. Alors, la sécurité dépend du calcul de $e^{ième}$ racine de g^e et r^e.
- Notre authentification est liée au paramètre secret r (identification) qu'on ne peut pas extraire, sauf si on calcule le $e^{ième}$ racine de r^e. En plus, elle est basée sur la vérification de m+r-r=m qui demande le calcul de $e^{ième}$ racine de $(m+r)^e$.

Déduction

Notre authentification a le même niveau que la signature de Shamir, puisque les deux sont semblables au niveau sécurité. En effet, la signature de Shamir est liée au calcul de $e^{ième}$ racine des deux paramètres, notre authentification est liée aussi à $e^{ième}$ racine des deux paramètres, et on ajoute une identification r et une intégrité de donné : m + r - r = message cohérent. Nous savons très bien que cette dernière est une condition principale dans la cryptographie (voir l'introduction).

Pour intégrer le RSA dans l'IBE seul l'utilisation des SEM est possible. Cette technique est typique, loin de la forme traditionnelle de RSA. Or, depuis une trentaine d'annés, la construction de RSA est liée à la factorisation. Dans la section qui suit nous proposons [4] une intégration de RSA basée sur cette forme, càd en conservant la proposition classique.

2.2.8 Troisième Version

Dans cette partie, nous utilisons les arguments suivants avec une conjonction de ce qu'on appelle OAEP.

- RSA **fort**. Un RSA est dit fort s'il a un nombre **premier** fort. Un nombre premier p est dit fort si $\frac{p-1}{2}$ est premier [51].
- Idée de RSA **multiprime** [102].

Contrairement à ce qu'on a vu précédemment de contacter la PKG par l'intermédiaire des SEM, dans cette partie on utilise plus cette technique et nous proposons :
La PKG publie un n fixe, ce n est sous la forme : n=(2p'+1)(2q'+1), p 'et q' sont deux nombres premiers et forts (principe de RSA fort) de 2048 - bits par exemple, en tenant en compte la sécurité actuelle (alors n est de 4098-bits).
Pour chaque récepteur qui demande de clé privée d_r, la PKG essaye de calculer cette dernière. Il apparaît que la PKG calcule d_r et il l'envoie simplement au récepteur concerné ; mais comment la PKG répond à tous les ID avec un n fixe ? !
Pour résoudre ceci, nous proposons qu'en plus de la génération de n, la PKG sélectionne pour chaque demande k, les nombres premiers $p_{i_{r_k}}$ et $q_{j_{r_k}}$ tels que :

$$e_{ID_{r_k}} \times d_{r_k} = 1 \, mod \, (p_{i_{r_k}} \times q_{j_{r_k}} \times \varphi(n)) \tag{2.45}$$

Le e_{ID_r} est calculé d'après notre méthode (partie 2.2.4)

Exactitude :

D'après le petit théorème de Fermat, nous avons :

$$\begin{cases} M^{p-1} = 1 \ mod \ p \\ M^{q-1} = 1 \ mod \ q \end{cases} \tag{2.46}$$

En utilisant le théorème des Restes Chinoix, nous avons :

$$M^{(p-1)(q-1)} = 1 \, mod \, n. \ Alors : \ M^{p_{j_{r_k}} . q_{j_{r_k}} . (p-1).(q-1)} = (1 \, mod \, n)^{p_{j_{r_k}} . q_{j_{r_k}}} = 1 \, mod \, n \tag{2.47}$$

Par conséquent :

$$M^{e_{ID_{r_k}} \times d_{r_k}} = M^{1+k' \times p_{i_{r_k}} \times q_{j_{r_k}} \times \varphi(n)} = M.(1 \, mod \, n) = M mod \, n \tag{2.48}$$

Méthode de génération avec une analyse de sécurité

Alice qui souhaite envoyer un message à Bob, contacte pour la première fois une autorité (PKG) pour extraire le paramètre n (qui est fixe) et les convenables paramètres pour générer e_{Bob}. Quand l'utilisateur Bob reçoit le message d'Alice chiffré par son identité, il doit contacter la PKG pour obtenir d_B.

Après avoir calculé $\varphi(n) = 4 \times p' \times q'$ par la PKG, par exemple pour 5 demandes, dans l'Extract, la PKG choisit cinq nombres premiers p_1, p_2, p_3, p_4, p_5 de 2048 -bits tels que :

Pour chaque e_{ID_i}, i $\in \{1, 2, 3, 4, 5\}$ (l'identité de Bob est l'une des cinq). La PKG calcule d_{ID_i} qui correspond à $\varphi'_i(n)$. Elle réserve cette dernière au convenable récepteur et elle la calcule comme suit :

$e_{ID_i} \times d_{ID_i}$ = 1 mod ($p_i \times p_{i+1} \times \varphi(n)$)= 1 mod $\varphi'_i(n)$, $\forall$ i $\in$ {1,2,3,4,5}. Alors nous avons :

$e_{ID_1} \times d_{ID_1}$ = 1 mod $(p_1 \times p_2 \times \varphi(n))$= 1 mod $\varphi'_1(n)$

$e_{ID_2} \times d_{ID_2}$ = 1 mod $(p_2 \times p_3 \times \varphi(n))$= 1 mod $\varphi'_2(n)$

$e_{ID_3} \times d_{ID_3}$ = 1 mod $(p_3 \times p_4 \times \varphi(n))$ = 1 mod $\varphi'_3(n)$

$e_{ID_4} \times d_{ID_4}$ = 1 mod $(p_4 \times p_5 \times \varphi(n))$= 1 mod $\varphi'_4(n)$

$e_{ID_5} \times d_{ID_5}$ = 1 mod $(p_5 \times p_1 \times \varphi(n))$ = 1 mod $\varphi'_5(n)$

Sous cette méthode, la PKG sert facilement avec un n fixe 5 demandes (la demande de Bob est parmi ces cinq).

Pour que cette méthode soit rigide contre les choix adaptatifs, elle doit vérifier les conditions suivantes :

1. $\forall i \in \{1, 2, 3, 4, 5\}$, $p_i \times p_{i+1}$, $p_i \times p'$, $p_{i+1} \times p'$, $p_i \times q'$, $p_{i+1} \times q'$ doivent être non factoriels (il est faisable puisque notre produit est de 4096-bits).
2. $\forall i$, $l(p_i) >> l(p_{i+1})$, avec par exemple $l(p_i) - l(p_{i+1})$ = 512 bits (l est la longueur). La différence signifie le poids ou encore la distance de Hamming et que $>>$ veut dire qu'il existe une différence.
3. $\forall \{i, j\} \in \{1, 2, 3, 4, 5\}$, $(p_i \times p_{i+1}) + (p_j \times p_{j+1})$ contient le nombre $p_{i,j}$ sachant que : $p_{i,j} \times p' \times q'$ doit être non factoriel.
4. $\forall i \in \{1, 2, 3, 4, 5\}$, $\varphi'_i(n)$ ne divise pas $\varphi'_{i+2}(n)$.

Ces choix peuvent-ils être effectifs ? La réponse est oui, en effet :

La première condition est destinée à n'extraire aucun des pi,p',q' à partir du produit. Alors que, la deuxième condition est consacrée pour résister aux choix adaptifs. Or, pour un bon choix de (k_i, k_j) avec lequel nous souhaitons attaquer $\varphi(n)$, p',q'... (pour (i,j) $\in$ {1,2,3,4,5}), nous montre que cette méthode de génération est peut être vulnérable aux choix adaptatifs. Mais en réalité cela ne peut pas se produire, comme l'exprime les cas suivants :

a) **Cas des i successifs**

$$k_i(p_i \times p_{i+1})\varphi(n) - k_{i+1}(p_{i+1} \times p_{i+2})\varphi(n) = p_{i+1}(k_i p_i - k_{i+1}p_{i+2})\varphi(n) = p_{i+1}\varphi(n) \tag{2.49}$$

On peut arriver à ceci puisque $p_i \bigwedge p_{i+2} = 1$; mais il faut choisir les bons (p_i , p_{i+2}) et pour le faire, c'est avec une probabilté $\frac{C^2_{n_p(2^{2048})}}{2^{2048}}$ (C est la combinaison). Choisir le bon (p_i , p_{i+2}) est lié à la condition : $|\, p_i \,|$ - $|\, p_{i+2} \,|$ = $|\, p_i \,|$ - $|\, p_{i+1} \,|$ + $|\, p_{i+1} \,|$ - $|\, p_{i+2} \,|$ = (512 + 512) bits = 1024 bits.

Alors cette probabilité est calculée sous la forme : P(B / A) = $\frac{P(A \bigcap B)}{P(A)} \simeq$ P(B). On doit négliger l'événement A par comparaison avec B comme il est simplement vérifié. Nous ne la prenons en considértion que lorsque nous obtenons les nombres premiers convenables p_i et p_{i+1}.

B est l'événement de choisir deux nombres premiers appropriés parmi les nombres de 2^{2048} bits. Alors que A est l'événement que deux nombres premiers doivent vérifier la condition 2.

Mais, même si nous atteindrons ce résultat càd $p_{i+1}\varphi(n)$, pour attaquer $\varphi(n)$ nous avons besoin de la probabilité : $\frac{c}{n_p(2^{2048})}$ où c est le nombre des choix pour attaquer p_{i+1}

(P($\bigcup_j p_j$)=$\sum_j P(p_j)$ = c, puisque $P(p_j)$ est la probabilité pour attaquer le nombre premier p_j, c'est une probabilité équiprobable).

Pour attaquer par exemple $\varphi'_i(n) = p_i p_{i+1}\varphi(n)$ (sachant que nous avons supposé qu'on a attaqué $p_{i+1}\varphi(n)$), on doit essayer avec la probabilité : $\frac{c'}{n_p(2^{2048})}$ (c'est la probabilité pour attaquer p_i)

Alors :

$$P(\varphi(n)/(k_i, k_{i+1})) \simeq \frac{\frac{c}{n_p(2^{2048})} \times \frac{(C^2_{n_p(2^{2048})})}{(2^{2048})}}{\frac{(C^2_{n_p(2^{2048})})}{(2^{2048})}} = \frac{c}{n_p(2^{2048})} \tag{2.50}$$

qui dépend de c ; mais comme p_{i+2} est inconnu cette probabilité sera négligeable.

Le terme $P(\varphi(n)/(k_i, k_{i+1}))$ signifie que l'événement $\varphi(n)$ est calculable, sachant que nous choisissons le vrai (k_i, k_{i+1}) ; mais le choix de ce dernier couple est après un bon choix de (p_i, p_{i+1}).

(A noter que : (k_i, k_{i+1}) est peut être calculé à partir de φ'_i et φ'_{i+1})

Comme conséquence de tout ceci, nous avons :

$$P(\varphi'_i(n)/(k_i, k_{i+1})) = P(\varphi'_i(n)/p_{i+1}\varphi(n)) \simeq \frac{\frac{c'}{n_p(2^{2048})} \times \frac{c}{n_p(2^{2048})}}{\frac{c}{n_p(2^{2048})}} = \frac{c'}{n_p(2^{2048})} \tag{2.51}$$

On peut dire que même si on attaque $\varphi(n)$ (ceci ne peut être attaqué qu'après des propres chois des p_j successives et suivant la précédente méthode des chois adaptatifs) pour attaquer l'un des $\varphi'_i(n)$ qui est plus important que $\varphi(n)$, il faut le faire avec une probabilité $\frac{n_p(2^{4096})}{2^{4096}} << 1$!!.
($\frac{n_p(2^{4096})}{2^{4096}}$ est la probabilité d'attaquer $p_j p_{j+1}$ dans $\varphi'_j(n)$ après avoir attaqué $\varphi(n)$)

b) **Cas des i non successifs**

Pour attaquer $\varphi(n)$ à partir de :

$$k_i(p_i \times p_{i+1})\varphi(n) - k_j(p_{j+1} \times p_{j+2})\varphi(n) = (k_i p_i \times p_{i+1} - k_j p_{j+1} \times p_{j+2})\varphi(n) \quad (2.52)$$

après un bon choix de (k_i, k_j) a une probabilité inférieure à : $\frac{2n_p(2^{2048})}{2^{2048}}$
Preuve de $\frac{2n_p(2^{2048})}{2^{2048}}$:
Par exemple pour $(p_1 \times p_2 - p_3 \times p_4)$ on a :
$| p_1 \times p_2 - p_3 \times p_4 | > || p_1 || p_2 | - | p_3 || p_4 ||$, puisque la relation $>$ est une relation d'ordre dans $\Re$, alors l'un des termes $| p_1 || p_2 |$,$| p_3 || p_4 |$ est plus grand que l'autre. Supposons que c'est $| p_1 | | p_2 |$, nous avons :
$|| p_1 || p_2 | - | p_3 || p_4 ||$ = $| p_1 || p_2 |$ - $| p_3 || p_4 |$
Alors : $| p_1 \times p_2$ - $p_3 \times p_4 | > | p_1 || p_2 |$ - $| p_3 || p_4 |$ et puisque $| p_1 | >$ - $| p_3 |$ ce qui donne :

$$| p_1 \times p_2 - p_3 \times p_4 | > | p_1 || p_2 | - | p_3 || p_4 | > - | p_3 || p_2 | - | p_3 || p_4 |$$
$$> - | p_3 | (| p_2 | + | p_4 |)$$

Si on applique la probabilité P à cette inégalité et comme ces événements sont uniformes et sûr, alors :

$$P(| p_1 \times p_2 - p_3 \times p_4 |) < P(- | p_3 | (| p_2 | + | p_4 |)) < P(- | p_3 |) + P(| p_2 | + | p_4 |)) \quad (2.53)$$

(puisque P(ab)≤ P(a)+P(b) d'après le fait que : $P(a \bigcap b) = P(ab) = P(a) + P(b) - P(a \bigcup b)$)
Comme nous l'avons signalé : $<$ est une relation d'ordre (nous sommes intéressés de comparer $| p_2 |$ et $| p_4 |$)). Alors par exemple nous avons :
P($| p_1 \times p_2 - p_3 \times p_4 |$) < P(-$| p_3 |$) + $P(2 | p_2 |)$ ≤ $\frac{2n_p(2^{2048})}{2^{2048}}$.
(calculer P(2 $| p_2 |$) est équivalent à calculer P($| p_2 |$)).

□

La troisième condition est a pour raison de ne pas attaquer $\varphi(n)$. Mais même si on attaque celle-ci, on doit attaquer $\varphi'_i(n)$; mais cela ne peut se faire qu'après une attaque de p_i, p_{i+1} ou de $p_i \times p_{i+1}$ qui ont respectivement une probabilité d'attaque : $\frac{n_p(2^{2048})}{2^{2048}}$ et $\frac{n_p(2^{4096})}{2^{4096}}$. On voit que les deux sont strictement inférieurs à 1 (càd qu'elles sont négligeables).

La quatrième condition a pour but : si $\varphi'_i(n)$ divise $\varphi'_{i+2}(n)$ (ou inversement) alors $\varphi'_i(n)$ = cte $\times$ $\varphi'_{i+2}(n)$ et donc $d_{ID_i} = cte \times d_{ID_{i+2}}$, ce qui est un risque.
L'autre cas qui est $\varphi'_i(n)$ divise $\varphi'_{i+1}(n)$ est exclu, or si nous l'avons, on peut avoir p_i divise p_{i+2} ce qui ne l'est pas, puisque p_{i+1} est un nombre premier.

Remarques 2.2.1 :

1. Le terme φ' n'est pas l'Indicateur d'Euler, c'est une notation seulement
2. Nous avons remplacé le terme **1** :longueur par || pour des raisons mathématiques
3. $n_p(x)$ est le nombre des nombres premiers qui existent dans [2 , x]
4. Les valeurs possibles pour des i non successives sont : $(p_1 \times p_2 - p_3 \times p_4)$, $(p_1 \times p_2 - p_4 \times p_5)$, $(p_2 \times p_3 - p_4 \times p_5)$.

Le $\varphi'(n)$ est le produit des quatre nombres premiers, avec un degré de sécurité de 8192-bits. Il convient bien le tableau suivant (tiré de [102]) qui prend en compte la sécurité des RSA multi-primes.

Module en bit	1024	2048	4096	8192
Nombres premiers	3	3	4	5

TABLE 2.1 – Nombre des nombres premiers convenables pour différents modules

Selon ce tableau pour un niveau de 8192-bits (on peut utiliser 4096-bits pour réduire la complexité), on peut intégrer cinq nombres premiers ; mais notre $\varphi'(n)$ est constitué seulement de quatre nombres premiers, on ne peut nul part le factoriser.
Avec ces cinq nombres premiers, nous servirons cinq demandes, puisque, trouver un nombre premier de 2^{2048} bits est coûteux, la question qui se pose est de savoir si on peut réutiliser les anciens nombres générés, effectivement la réponse est positive. En effet, par exemple avec p_1, on peut associer $p_1 p_i \varphi(n)$ sachant que $p_1 p_i$ satisfait la condition 2. Alors, pour i valeur de φ' et avec un p_1 on sert i identité. Comme estimation numérique, on peut utiliser avec ce p_1 tous les nombres p_i qui existent dans l'intervalle $[2^{512}, 2^{2048} - 2^{512}]$. La borne inférieure de cet intervalle est liée au fait que si nous voulons choisir un nombre multi-prime de quatre nombres, il est préférable de passer 512 bits. Alors que la borne supérieure est basée sur la condition 2.
Ce principe est valable pour les autres p_i (recyclage d'utilisation).

2.2.9 Étude de sécurité en général

Comme nous l'avons signalé, intégrer le RSA (plus particulièrement le RSA-OAEP) dans l'IBE souffre de plusieurs problèmes. Le premier qui intervient est comment lier l'exposant e avec l'identité. En [28][64] les auteurs ont mis à profit quelques solutions en se basant

sur ce qu'on appelle Random Oracle. Malheureusement, leurs propositions portent des faiblesses, notamment, au niveau de la précision de la primalité entre l'exposant construit et la fonction d'Euler $\varphi(n)$ utilisée, ceci revient au comportement aléatoire et incontrôlable du random oracle qui rentre en jeu. A priori, préciser la primalité entre l'exposant e et la fonction d'Euler $\varphi(n)$ (convenable au module n) est une condition principale pour le RSA et que toute irréalisation de cette condition peut bloquer son fonctionnement. En plus de cela, les propositions [28][64] peuvent être victimes simples de ce qu'on appelle malléabilité, qui peut donner la puissance à un adversaire pour calculer le chiffré d'une convenable entité, après avoir accéder à quelques chiffrements. Nous avons pris en considération ces faiblesses, nous avons proposé une méthode conduite à un e_{ID} premier (partie 2.2.4) et donc automatiquement premier avec $\varphi(n)$. Le fait d'avoir un e_{ID} premier contourne bien le problème de malléabilité puisqu'on ne peut pas avoir la linéarité entre les e_{ID_i}, $\forall i$. En plus et au contraire de [28][64], notre méthode est sous le modèle Satandard qui est un modèle sûr et bien adapté à la vérification, cela est bien avantageux contre la malléabilité.

Le problème d'intégrer le RSA dans l'IBE n'est pas limité à ce qu'on a annoncé. Privilèger de lier l'exposant e avec l'identité comme nous avons vu dans la partie 2.2.4, pose la question sur le module n convenable et sur le fait de le changer après chaque utilisation. Fixer le module pour tous les utilisateurs est un souhait désirable, puisqu'il conserve le cadre dont lequel l'IBE a été tracé (réduire l'utilisation du certificat). Mais participer un seul n entre un nombre i des utilisateurs (dans une organisation) peut factoriser ce n après l'avoir reçu des d_{ID_i}. Pour cette raison, les auteurs dans [28][64] ont utilisé ce qu'on appelle SEM dont le but est de diviser la clé privée sur deux, une donnée à l'utilisateur alors que l'autre reste chez l'autorité de confiance. Alors l'utilisateur ne peut pas accéder à une demi-clé laissée chez le SEM. Effectivement ceci nécessite un confiant SEM, ce qui pose la question concernant la sûreté de cette méthode. Outre que cela, les méthodes proposées dans [28][64] demandent beaucoup d'authenticité, les utilisateurs doivent être confiants (puisqu'il peut exister des multiutilisateurs qui peuvent utiliser l'aide du SEM pour attaquer le challenge), elles sont lourdes, elles ne conservent pas le principe du RSA classique.

Dans la nouvelle méthode que nous avons proposé lors de ce chapitre (partie 2.2.8) nous avons rectifié ces faiblesses : premièrement nous avons réservé le principe du RSA classique, dans notre méthode on n'a pas utilisé le SEM, les utilisateurs demandés sont tous indépendants, ils dépendent seulement du PKG. Prenant en compte tout cela, nous pouvons confirmer que : La troisième version jointe de notre méthode qui génère le e_{ID} est sécurisée dans le sens Semantic-CCA2 (nous notons que Semantic-ATK est équivalent à IND-ATK, d'après [37]), puisque :

1. Le RSA-OAEP est sécurisé par [76]
2. La méthode pour générer e pour un ID résiste à la malléabilité
3. Le e_{ID} construit est standard (on n'a utilisé aucun random)
4. Tous les utilisateurs sont indépendants

En utilisant cela joigne du lemme suivant rend la troisième version Semantic-CCA2 sécurisée

Lemme 2.2.4 : *Le IBE-RSA/OAEP système muni de n utilisateurs est semantic sécurisé avec :*

$$Succ_n{}^{IBE-RSA}(t_n, q_d, q_e) = Succ_n{}^{RSA}(t_n, q_d, q_e) \leq q_e\, n\, Succ_1{}^{IBE-RSA}(t_1, q_d)$$
$$= q_e\, n\, Succ_1{}^{RSA}(t_1, q_d) \quad \text{où} \quad t_1 = t_n + O(log(q_e n))$$

Nous avons besoin de ce lemme puisqu'un seul n est divisé sur plusieurs identités.
A noter que dans ce lemme nous n'avons pas besoin d'un confiant utilisateur (contrairement à [64]). Puisque nous n'avons pas utilisé le SEM et que les utilisateurs sont indépendants (nous avons étudié cela dans la partie : i successives et non successives)

2.3 Conclusion

Le présent chapitre est réservé aux cryptoytèmes sans couplage, nous l'avons abordé par l'examen d'une proposition dans CT-RSA 2009 qui traite l'anonymat du schéma de Cocks. Nous avons mis en défaut ce travail en proposant une alternative efficace, notre procédure conserve la même longueur que Cocks, en plus, elle garantie l'anonymat. Nous pouvons concrétiser cela d'après le tableau suivant :

	Params	Encrypt	Decrypt	Anonymat
Schéma [10]	2m+2	2Additions+ 2Multiplications+ 2Inverses+ ($k_1 + k_2$)Symboles de Jacobi	$\geq k_1$ ou k_2 Symboles de Jacobi	non sûr
Schéma varaint de Cocks	2	2Additions+ 2Multiplications+ 2 Symboles de Jacobi	1 Symboles de Jacobi	sûr

TABLE 2.2 – Calcul de complexité des schémas : [10] et le schéma variant de Cocks.

Les paramètres m, k_1 et k_2 sont d'après le papier [10].
Ensuite, nous avons examiné l'intégration de RSA dans l'IBE. Pour le faire, nous avons exposé trois versions jointes d'une proposition efficace pour un exposant lié à l'identité. Les deux premières développent des idées dans la littérature qui utilisent SEM, alors que la troisième suit le RSA classique, cette version conserve la forme traditionnelle de RSA et elle résoud le problème IB-RSA.

Chapitre 3

Cryptosystèmes avec Couplage : nouvelles propositions

Dans le chapitre 2, nous avons exposé deux crypto systèmes, le Cocks et le RSA. Nous avons examiné l'anonymat du premier en mettant par défaut la proposition [10]. Rendre ce schéma anonyme augmente le degré de sa sécurité. Alors que pour le deuxième nous avons tenté la question laissée sans réponse absolue par Shamir en 1984, et qui concerne l'intégration de RSA dans l'IBE. Intégrer cette primitive sans conteste dans l'évolutive technologie IBE est une valeur marquée pour cette dernière. L'avantage de ces deux schémas est qu'ils ne nécessitent aucune utilisation de couplage. Le contexte couplage sollicite un examen plus approfondi, son implémentation demande beaucoup d'opérations, elle est ainsi coûteuse. C'est ce que nous examinerons dans le chapitre 5.

Le présent chapitre consiste d'enrichir l'IBE par de nouveaux schémas avec couplages, en effet, les propositions dédiées dans ce cadre sont peu nombreuses et présentent ainsi une faiblesse mature pour l'IBE. La difficulté de proposer des nouveaux schémas basés sur les couplages se présente dans la satisfaction de construire une clé privée qui relie le ID et la clé maîtresse par la raideur de celle-ci contre les attaques internes et externes. En plus, il faut faire attention à la rigidité du problème que présente la clé privée construite, celui-ci représente la mesure intuitive du niveau de sécurité du schéma.

Dans la première et deuxième parties du chapitre, on propose deux schémas dans le modèle sélective ID, le premier est un IBE alors que le deuxième est un HIBE, l'étude de sécurité de chaque schéma et les comparaisons avec les propositions du précède sont extrêmement cités et discuter.

Dans la troisième partie, on exploite un quatrième schéma pour l'IBE sous le modèle Random Oracle, la sécurité de la version complète de ce schéma est entièrement étudiée dans le modèle

des Randoms Oracles, repose sur le problème 4-EBDHP qui est bien rigide devant l'attaque de Cheon, une partie qui argumente la performance du schéma est bien réservée.

3.1 Schéma IBE compétitive à BB1 et BB2 dans le modèle sélective ID

Un schéma à clé publique (plus particulièrement le chiffrement basé sur l'identité) est dit sûr s'il est prouvé sécurisé contre les études de simulations, celles-ci tiennent en compte le but de l'attaqueur et le modèle utilisé. Parmi les buts il y a indistinguishabilité IND et sémantique but, concernant les modèles, nous citons CPA, CCA. Leur mariage donne IND-CPA, sémantique-CPA, sémantique-CCA, IND-CCA. L'étude considérée comme forte et concrétisant pour une sécurité idéale est IND-CCA, pour l'IBE nous parlons de IND-ID-CCA (aussi : sémantique-ID-CPA, sémantique-ID-CCA, IND-ID-CPA). Ce IND-ID-CCA (ainsi que les autres) appartient à un domaine complet (full domain) dont lequel l'identité visée à attaquer est déclarée dans le challenge. Un type dit faible est introduit en 2003 par Canetti et al [45] il s'appelle sélective-ID. Dans ce modèle, l'identité cible à être attaqué est déclarée dans le début, càd dans le Setup. Il est prouvé dans [23] que le passage de sélective ID à un domaine complet nécessite une multiplication par N (produit une dégradation dans la sécurité). Malgré cette faiblesse, en Eurocrypt 2004 Boneh et Boyen [23] ont proposé deux crypto systèmes sous ce type, ce sont les seuls connus dans la littérature tracée dans le même type. Le premier est un HIBE basé sur le problème DBDHP (Decisional of Bilinear Diffie and Hellman Problem) et sous l'approche de Commutative Blinding, il est connu par BB1. Alors que le deuxième est un IBE sous l'Exponent-Inversion approche nommé BB2, il est fondé sur Dq-BDHIP (Decisional q-Invertible of Bilinear Diffie and Hellman Problem). En combinant l'idée de l'inverse utilisée dans BB2 et restant dans l'approche des Commutative Blinding, nous allons exposer notre IBE schéma [5] qui sera plus réduit que BB1 (version IBE [25]) et même que BB2.

3.1.1 Notion d'identité sélective (sélective-ID) pour IBE/HIBE

Le fonctionnement de sélective-ID est selon les algorithmes déclarés ci dessous, ici, nous donnons la version CPA, càd sans utiliser l'extraction des demandes du déchiffrement dans la phase 1. Nous donnons la définition dans le cas d'un IBE et il est facile de la généraliser pour un HIBE.

Init :

Un adversaire A relève le défi : l'identité ID^*.

Setup :

Le challenger déroule le Setup algorithme. Il donne à l'adversaire le système des paramètres résultants dans le params et il garde la clé maîtresse.

Phase 1 :

L'adversaire issu des demandes $q_1, ..., q_m$ avec q_i est :

— Demande de la clé privée pour un $< ID_i >$ tel que : $ID_i \neq ID^*$ et, ID_i n'est pas le préfixe de ID^*. Le challenger répond en déroulant l'algorithme KeyGen (ou bien Extract-voir chapitre 1) pour générer la clé privée d_i correspondante à la clé publique de $< ID_i >$. Il envoi d_i à l'adversaire.

Challenge :

Une fois l'adversaire décide de terminer la Phase 1, il fait sortir deux plaintexts $m_0, m_1 \in \mathtt{M}$ de même longeur. Le challenger sélectionne un bit arbitraire b $\in \{0, 1\}$, il calcule le ciphertext C = Encrypt(params, ID^*, m_b). Puis il l'envoie comme un challenge à l'adversaire.

Phase 2 :

Comme Phase 1

Guess :

Finalement, l'adversaire fait sortir un guess (estimation) $b_0 \in \{0, 1\}$. Il gagne si $b = b_0$.

Nous référons A comme un IND-sID-CPA, son avantage pour attaquer un schéma est $Adv_{\varepsilon,A}$ = | Pr[b = b_0] - $\frac{1}{2}$ |, c'est une probabilité d'un bit de gain construit arbitrairement entre le challenger et l'adversaire.

Nous disons qu'un IBE (pour le HIBE de niveau k, le ID^* se refère à : ${ID_1}^*, {ID_2}^*, ..., {ID_k}^*$) d'un système E est (t, q_{ID}, ε)-sélective-identité et adaptivement sécurisé, si, pour chaque IND-sID-CPA adversaire A qui se déroule dans un temps t, qui fait au moins q_{ID} demandes de clés privées qu'il choisit, on a :

$$Adv_{\varepsilon,A} = |Pr[b = b_0] - \frac{1}{2}| \ < \ \varepsilon \tag{3.1}$$

3.1.2 Éstimation de quelques problèmes bilinéaires de Diffie Hellman

Fixant les paramètres : G_1, G_1 et G_T ; ainsi que ê, tels que :

— G_1, G_2 et G_T des groupes cycliques d'ordres premier p.

— g un générateur de G_1 ou de G_2.

— ê : $G_i \times G_i \longrightarrow G_T$ où i $\in \{1,2\}$, une application bilinèaire sous forme couplage.

Définition 3.1.1 *: (Décisional Bilinèaire Diffie-Hellman Problème (DBDHP). Soit g un générateur de G_1. Le DBDHP dans $< G_1, G_T$, ê $>$ est alors :*
Étant donné $<$ g, g^a, g^b, g^c, z $>$ pour a, b, c $\in Z_q$ et T $\in G_T$. Nous disons qu'un algorithme A à un avantage ε pour résoudre le décisionnel BDHP dans G_T si :

$$|Pr[g, g^a, g^b, g^c, \hat{e}(g,g)^{abc}] - Pr[g, g^a, g^b, g^c, T]| > \varepsilon$$

Cette probabilité est après un choix arbitraire de : un générateur g dans G_1, (a, b, c) $\in Z_q \times Z_q \times Z_q$, T $\in G_T$ un bit arbitraire choisit par A. La distribution à droite est référenciée par P_{BDHP} tandis que celle à gauche est référenciée par R_{BDHP}.

Définition 3.1.2 : *((Decisional)k-Bilinear Diffie Hellman Inversion Problem (Dk-BDHIP)). Soit $g \in G_2^*$ (ou dans G_1^*). Peut-on arriver à l'inégalité suivante :*

$$|Pr[g, g^x, g^{x^2}, ..., g^{x^k}, \hat{e}(g,g)^{\frac{1}{x}}] - Pr[g, g^x, g^{x^2}, ..., g^{x^k}, T]| > \varepsilon$$

Pour un $g^x, g^{x^2}, ..., g^{x^k}$ et T donnés (où T est dans G_T).

3.1.3 Schéma sous forme d'IBE

Comme nous l'avons signalé, le modèle sélective ID est une notion faible. Le lecteur peut se reporter à [87] pour avoir une idée plus vaste sur le poids de ce modèle. Il est aussi usuel que le schéma BB1 tracé sous ce modèle est plus complexe, ce qui perd l'efficacité pour ce schéma. Dans le travail [5], nous avons pensé à un schéma plus réduit sous sélective ID, pour le faire, nous avons combiné le principe de l'inverse utilisé dans l'Extract du schéma BB2 et l'approche du commutative Blinding d'où il est construit BB1.

Premier schéma : nouveau IBE schéma

Setup : Fixant un paramètre de sécurité t. Soit (G_1, G_T) deux groupes bilinéaires.
Choisir un générateur $g \in G_1$ et soit $P_{pub_1} = g^l \in G_1{}^\star$.
Calculer : $e(g,g) = x$ et $e(g,g)^a = x^a = y$ (avec e représente le couplage)
Les paramètres publiques sont : M_{pk}= $\{G_1, G_T, P_{pub_1}$, x, y $\}$.
La clé maîtresse est M_{sk}= {l,a}
L'espace du message est : $\{0,1\}^n$; l'espace du ciphertext est : $G_1{}^* \times \{0,1\}^n$.
Extract : Étant donné une identité $ID_A \in \{0,1\}^n$ d'une entité A, M_{pk} et M_{sk}.
Sélectionner un $r_{ID_A} \in Z_q$, puis retourner $g^{\frac{a+ID_A}{r_{ID_A}\, l}}$=$g^{\frac{\frac{a}{r_{ID_A}}+r'_{ID_A}\, ID_A}{l}}$=$g^{\frac{a'+r'_{ID_A}\, ID_A}{l}}$.
Alors :

$$d_{ID_A} \quad = \quad (r_{ID_A}, g^{\frac{a+ID_A}{r_{ID_A}\, l}}) = (r_{ID_A}, d_A)$$

Encrypt : Étant donné m $\in \mathbb{M}$, ID_A et M_{pk}, Suivre les étapes :

1. Choisir un s arbitraire dans Z_q
2. Calculer : $e(g,g)^{s(ID_A+a)} = (x^{ID_A}y)^s$

Le ciphertext est : $\mathbb{C} = (g^{ls} = {P_{pub_1}}^s, m.e(g,g)^{s(ID_A+a)})$=(u,v).

Decrypt : Étant donné le ciphertext $\mathbb{C}$ = (u,v), ID_A, d_A et M_{pk}.

Le déchiffrement de $\mathbb{C}$ est d'après :

1. Calculer e$(u^{rID_A}, e(u^{rID_A}, g^{\frac{a+ID_A}{rID_A{}^l}}))$, ensuite, faire sortir le m=$\frac{v}{e(u^{rID_A}, g^{\frac{a+ID_A}{rID_A{}^l}})}$

Remarque 3.1.1 : *Le paramètre de sécurité t doit satisfaire les recommandations du NIST, ECRYPT ou autres. Remplir le niveau souhaité demande de faire attention au plus grand paramètre qui construit la factorisation de l'ordre de la courbe adaptée au calcul du couplage e (voir plus tard le chapitre 5, partie 5.6)*

Exactitude

Puisque :

$$e(u^{rID_A}, g^{\frac{a+ID_A}{rID_A{}^l}})) = e(g^{lsrID_A}, g^{\frac{a+ID_A}{rID_A{}^l}}) = e(g,g)^{s(ID_A+a)}$$

Le nouveau IBE schéma est alors correct.

3.1.4 Preuve de sécurité sous le modèle sélective ID du nouveau IBE schéma

Avant de prouver la sécurité du nouveau IBE schéma, nous notons que Dk^--BDHIP signifie qu'au sens de la définition 3.1.2 on utilise n'importe quel k > 0, ce dernier paramètre n'est pas lié au nombre d'utilisateurs comme avec Dk-BDHIP ([23]), il est plutôt de notre choix. On peut le choisir 2 ou n'importe quel nombre, par contre avec Dk-BDHIP on a besoin d'au moins 2^{50} d'après [118] pour un niveau de sécurité égal à 80-bits (niveau de sécurité dans le cas de la cryptographie symétrique).

La sécurité du nouveau IBE schéma est basée sur la rigidité de Dk^--BDHI, d'après :

Théorème 3.1.1 : *Supposons que (t, k^-, ε)-Décision BDHI est rigide dans un groupe cyclique G_1 de longueur p ($|G_1|$ = p). Alors le nouveau IBE schéma est (t', k_S, ε)-sélective identité, il est chosen plaintext (IND-sID-CPA) sécurisé, avec un avantage :*

$$adv^{nouveau\ IBE\ schéma}(t') > adv^{Dk^- - DBDHIP}(t - O(\tau k^-)) \quad \text{pour chaque} \quad k_S\ (< k^-).$$

Où τ est le temps nécessaire pour calculer l'exponentiation dans l'étude suivante :

Preuve :

Supposons qu'un adversaire A à un avantage ε pour attaquer le nouveau IBE schéma. Nous construisons un algorithme B qui utilise A pour résoudre le problème Décision k^--BDHI dans G_1. L'algorithme B reçoit comme entrées : des (k^-+2)-paramètres arbitraires $(g, g^{\alpha}, g^{\alpha^2}, ..g^{\alpha^{k^-}}, T)$ $\in G_1^{k^-+1} \times G_T$ qui soient extraites à partir de P_{BDHI} (avec $T = e(g,g)^{1/\alpha}$) ou de R_{BDHI} (avec T est uniforme et indépendante dans G_T : groupe d'arrivée du couplage). Le but de l'algorithme B est de faire sortir 1 si $T = e(g,g)^{1/\alpha}$ et 0 sinon. L'algorithme B travaille en collaboration avec A pour obtenir un gain sous le modèle sélective-ID comme suit :

Setup :

Pour générer le système des paramètres, l'algorithme B fait ce qui suit :

Au début, l'algorithme A donne à B l'identité I^*=$\frac{a_1}{b_1}$ où il veut attaquer. Le gain de l'identité sélective commence, mais l'algorithme B a besoin de l'étape de préparation suivante :

Étape de préparation

Dans l'étape de préparation, l'algorithme B choisit un arbitraire x, puis, il calcul $b_1 x$. Après, il calcule implicitement :

$$f(\alpha) = \sum_{i=1}^{k^-} c_i \alpha^i \tag{3.2}$$

Il choisit arbitrairement r_0, il calcule aussi et implicitement

$$r_1 = r_0 \sum_{i=1}^{k^-} c_i \alpha^{i-1} \tag{3.3}$$

En fin, il calcule h=$g^{f(\alpha)}$, il publie ce h.

Phase 1 :

A issu au plus k_S demande de clé privée, avec $k_S < k^-$. Considérons la i ème demande pour une clé privée correspondante à la clé ID_i telle que : (I_i =) $ID_i \neq ID^*$ (=I^*).

On a besoin des réponses de clés privées sous la forme (r, $h^{\frac{a+r(I_i - I^*)}{\alpha}}$).

Le I_i représente une identité générale qu'on a fixé et I^* représente l'identité qu'il faut

attaquer (identité sous forme défie). r est uniformément distribué dans Z_p.

Algorithme B répond aux demandes comme suit :

Premièrement, il est possible que la clé privée dans le nouveau IBE schéma peut avoir un syntaxe sous la forme : $g^{\frac{a+rID_A}{l}}$ au lieu de celui $g^{\frac{a+ID_A}{rl}}$, puisque :

$$g^{\frac{a+ID_A}{rl}} = g^{\frac{a}{rl}+\frac{r'ID_A}{l}} = g^{\frac{a'+r'ID_A}{l}} \tag{3.4}$$

Nous avons besoin de ce dernier pour simplifier la preuve.

B pose R=$\frac{x}{r_0} + r_1$, il calcule implicitement :

$$\begin{aligned}
R &= \frac{f(\alpha)}{f(\alpha)}\left(\frac{x}{r_0} + \frac{r_1}{I_i - I^*} I_i - I^*\right) \\
&= \frac{f(\alpha)}{\alpha \sum_{i=1}^{k^-} c_i \alpha^{i-1}}\left(\frac{x}{r_0} + \frac{r_1}{I_i - I^*}(I_i - I^*)\right) \\
&= \frac{f(\alpha)}{\alpha}\left(\frac{x}{r_0 \sum_{i=1}^{k^-} c_i \alpha^{i-1}} + \frac{r_1}{\sum_{i=1}^{k^-} c_i \alpha^{i-1}(I_i - I^*)}(I_i - I^*)\right) \\
&= \frac{f(\alpha)}{\alpha}\left(\frac{x}{r_0 \sum_{i=1}^{k^-} c_i \alpha^{i-1}} + \frac{r_0 \sum_{i=1}^{k^-} c_i \alpha^{i-1}}{\sum_{i=1}^{k^-} c_i \alpha^{i-1}(I_i - I^*)}(I_i - I^*)\right) \\
&= \frac{f(\alpha)}{\alpha}\left(\frac{x}{r_0 \sum_{i=1}^{k^-} c_i \alpha^{i-1}} + \frac{r_0}{I_i - I^*}(I_i - I^*)\right) \\
&= \frac{f(\alpha)}{\alpha}\left(a' + r'(I_i - I^*)\right)
\end{aligned}$$

Avec r'=$\frac{r_0}{I_i - I^*}$ qui peut être calculé facilement par B.

Le a'=$\frac{x}{r_0 \sum_{i=1}^{k^-} c_i \alpha^{i-1}}$ est la clé maîtresse, qui est non connue par B, c'est comme α.

Remarque 3.1.2 : *, A peut publier g^a dans un système de paramètres. Pour éviter le calcul de a, B peut bien choisir son x de telle manière que : $g^{a'} = g^{\frac{x}{r_0 \sum_{i=1}^{k^-} c_i \alpha^{i-1}}}$ soit calculable (il suffit que x=$\sum_{i=1}^{k^-} c_i \alpha^{i-1}$). Après, B cherche un σ tel que : $g^a g^\sigma = g^{a'}$.*

Alors B peut calculer facilement g^R comme il sait $g^{\frac{x}{r_0}}$ et g^{r_1}.

Mais :

$$g^R = g^{\frac{f(\alpha)}{\alpha}(a' + r'(I_i - I^*))} = h^{\frac{a' + r'(I_i - I^*)}{\alpha}} \tag{3.5}$$

Qui est une clé privée valide, et alors B peut donner à A la clé privée (r',$h^{\frac{a'+r'(I_i-I^*)}{\alpha}}$).

Avec, B n'a pas l'avantage de calculer une clé privée pour I^*.

Challenge :

A fait sortir deux messages $M_0, M_1 \in G_1$. L'algorithme B sélectionne un bit arbitraire b $\in \{0,1\}$ et un l' $\in {Z_p}^*$ arbitraire aussi. Il répond avec un ciphertext préparé comme suit :

On a : $h^s = h^{\frac{s}{\alpha}\cdot\alpha} = h^{l'\alpha}$ = c_1, avec l'=$\frac{s}{\alpha}$

Et que :

$$c_2 = MT_h^{\frac{s(xb_1+a_1}{b_1})} = T_h^{s(x+I^*)} \quad \text{(ou plutôt } \quad c_2 = MT_h^{\frac{s(ab_1+a_1}{b_1})} = T_h^{s(a+I^*)})$$

Alors si $T_h = e(h,h)^{\frac{1}{\alpha}}$, nous avons :

$$e(h,h)^{\frac{s}{\alpha}(x+I^*)} = c_2 = e(h,h)^{l'(x+I^*)} \tag{3.6}$$

La combinaison CT=$(c_1, c_2) = (h^{l'\alpha}, e(h,h)^{l'(x+I^*)})$ est un ciphertext valide sous ID^*.

Si T_h est uniforme dans G_1, alors CT est indépendant du bit b.

Phase 2 :

A issu plus de demandes de clés privées, avec un total d'au plus $k_S < k^-$. L'algorithme B répond comme avant (càd dans la phase 1).

Guess :

Finalement, A fait sortir un guess (estimation) b' $\in \{0, 1\}$. Si b = b' alors B fait sortir 1 qui signifie que T = $e(g,g)^{\frac{1}{\alpha}}$. Sinon, il fait sortir 0 qui signifie que T $\neq e(g,g)^{\frac{1}{\alpha}}$.

Lorsque l'entrée de type $k^- + 2$ est calculée à partir de P_{BDHIP} (où T = $e(g,g)^{\frac{1}{\alpha}}$) alors l'opinion de A est identique à son opinion dans l'attaque réelle et donc A doit satisfaire :
|Pr[b = b'] - 1/2| $> \varepsilon$. D'une autre part, lorsque l'entrée de type $k^- + 2$ est calculée à partir de R_{BDHIP} (ou T est uniforme dans G_T), ce qui donne Pr[b = b'] = 1/2. Alors, avec g est uniforme dans G_1 et T est uniforme dans G_T, nous avons alors :

$$|Pr[g, g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^{k^-}}, \hat{e}(g,g)^{\frac{1}{\alpha}}] - Pr[g, g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^{k^-}}, T]| \geq |(\frac{1}{2} \pm \varepsilon) - \frac{1}{2} = \varepsilon| \tag{3.7}$$

3.1.5 Première discussion

• **Calcul de complexité pour BB1, BB2 et le nouveau IBE schéma**

	BB1 (version IBE)
Params	$2Exp_{ffi_{G_1/Z_q}}$+1coup+$1Exp_{ffi_{G_T/Z_q}}$
Extract	$2Mul_{ffi_{Z_q/Z_q}} + 2Exp_{ffi_{G_1/Z_q}}$
Encrypt	$1Mul_{ffi_{Z_q/Z_q}}$+$3Exp_{G_1/Z_q}$+$1Exp_{ffi_{G_T/Z_q}}$
Decrypt	2coup+$1Div_{ffi_{G_T/G_T}}$
Somme	3coup+$1Div_{ffi_{G_T/G_T}} + 3Mul_{ffi_{G_1/G_1}} + 7Exp_{ffi_{G_1/Z_q}} + 2Exp_{ffi_{G_T/Z_q}}$

TABLE 3.1 – Complexité de BB1

	BB2
Params	$2Exp_{ffi_{G_1/Z_q}}$+1coup
Extract	$1Mul_{ffi_{Z_q/Z_q}} + 1Inv_{ffi_{Z_q/Z_q}} + 1Exp_{ffi_{G_1/Z_q}}$
Encrypt	$1Mul_{ffi_{Z_q/Z_q}}$+$3Exp_{ffi_{G_1/Z_q}}$+$1Exp_{ffi_{G_T/Z_q}}$+$1Mul_{ffi_{G_1/G_1}}$
Decrypt	1coup+$1Div_{ffi_{G_T/G_T}} + 1Mul_{ffi_{G_1/G_1}} + 1Exp_{ffi_{G_1/Z_q}}$
Somme	2coup+$1Div_{ffi_{G_T/G_T}} + 2Mul_{ffi_{G_1/G_1}} + 7Exp_{ffi_{G_1/Z_q}}$+$1Inv_{ffi_{Z_q/Z_q}} + 2Mul_{ffi_{G_1/Z_q}}$

TABLE 3.2 – Complexité de BB2

	Nouveau IBE schéma
Params	$1Exp_{ffi_{G_1/Z_q}}$+1coup+$1Exp_{ffiG_T/Z_q}$
Extract	$1Exp_{ffi_{G_1/Z_q}}$+$2Mul_{ffi_{Z_q/Z_q}}$+$1Inv_{ffi_{Z_q/Z_q}}$
Encrypt	$1Mul_{ffi_{G_T/G_T}}$+$2Exp_{ffi_{G_T/Z_q}}$+$1Exp_{G_1/Z_q}$
Decrypt	1coup+$1Div_{ffi_{G_T/G_T}}$+$1Exp_{ffi_{G_1/Z_q}}$
Somme	2coup+$1Div_{ffi_{G_T/G_T}} + 2Mul_{ffi_{Z_q/Z_q}} + 1Mul_{ffi_{G_T/G_T}}$+ $3Exp_{ffi_{G_1/Z_q}}$+$3Exp_{ffi_{G_T/Z_q}}$+ $1Inv_{ffi_{Z_q/Z_q}}$

TABLE 3.3 – Complexité du nouveau IBE schéma

Les notations utilisées dans les tableaux 3.1, 3.2 et 3.3 signifient :
Exp_{ffi_G} : Multiplication Scalaire ; Exp_{ffi} : Exponentiation dans un corps fini ; Inv_{ffi} : Inversion dans un corps fini ; Mul_{ffi} : Multiplication dans un corps fini ; coup : couplage. Div : Division
En plus, on a : $Exp_{ffi_{*/**}}$, par exemple, (la même chose sera dite pour les autres opérations) signifie l'Exponentiation d'un corps fini regroupé dans */**, le * est la base de l'exponentiation, tandis que ** représente la base de l'exposant.

• **Test d'efficacité**

Complexité(BB1-version IBE) - Complexité(Nouveau IBE schéma)

$$\begin{aligned}
&= (3coup + 1Div_{ffi_{G_T/G_T}} + 3Mul_{ffi_{G_1/G_1}} + 7Exp_{ffi_{G_1/Z_q}} + 2Exp_{ffi_{G_T/Z_q}}) \\
&\quad - (2coup + 1Div_{ffi_{G_T/G_T}} + 2Mul_{ffi_{Z_q/Z_q}} + 1Mul_{ffi_{G_T/G_T}} + 3Exp_{ffi_{G_1/Z_q}} \\
&\quad + 3Exp_{ffi_{G_T/Z_q}} + 1Inv_{ffi_{Z_q/Z_q}}) \\
&\quad = 1coup + 4Exp_{ffi_{G_1/Z_q}} + 3Mul_{ffi_{G_1/G_1}} - 1Inv_{ffi_{Z_q/Z_q}} - 2Mul_{ffi_{Z_q/Z_q}} \\
&\quad - 1Mul_{ffi_{G_T/G_T}} - 1Exp_{ffiG_T/Z_q} >> 0
\end{aligned}$$

Et que :

Complexité(BB2) - Complexité(Nouveau IBE schéma)

$$\begin{aligned}
&= (2coup + 1Div_{ffi_{G_T/G_T}} + 2Mul_{ffi_{G_1/G_1}} + 1Exp_{ffiG_T/Z_q}+7Exp_{ffi_{G_1/Z_q}} \\
&+ \ 1Inv_{ffi_{Z_q/Z_q}} + 2Mul_{ffi_{G_1/Z_q}}) \text{ - } (2coup + 1Div_{ffi_{G_T/G_T}} + 2Mul_{ffi_{Z_q/Z_q}} \\
&+ \quad 1Mul_{ffi_{G_T/G_T}} + 3Exp_{ffi_{G_1/Z_q}} + 3Exp_{ffi_{G_T/Z_q}} + 1Inv_{ffi_{Z_q/Z_q}}) \\
&= 4Exp_{ffi_{G_1/Z_q}} + 1Mul_{ffi_{G_1/G_1}} + 2Mul_{ffi_{G_1/Z_q}} - 2Mul_{ffi_{Z_q/Z_q}} \\
&- 2Exp_{ffi_{G_T/Z_q}} >> 0
\end{aligned}$$

Conclusion

Le nouveau IBE schéma est plus efficace au niveau complexité que BB1 (version IBE) et il est un peu proche de BB2 (même efficace que lui). Ce dernier est basé dans ses études de simulations sur Dk-BDHIP, avec k est lié à l'identité demandée. Par contre, le nouveau IBE schéma est basé sur Dk^--BDHIP, où $k^- <<$ k. Alors le nouveau IBE schéma est aussi plus efficace que BB2 selon les résultats de Cheon[52] qui trouve que la complexité d'attaque des problèmes : k choisi-type de problème sous forme exponentiation, est : O(p/(le k choisi)).

3.2 Schéma HIBE compétitive à BBG dans le modèle sélective$^+$ ID

A Eurocrypt 2005, Boneh, Boyen et Goh [30] ont proposé un schéma qui réduit le ciphertext d'un HIBE de k paramètres à seulement trois et le Decrypt de k produit de couplages à seulement deux, la réduction est fixe quel que soit le ciphertext. Malheureusement, dans [30] il est nécessaire que l'identité qui représente le challenge soit dans Z_q^*, ce qui limite la sélection des identités lors des preuves de sécurité. En plus, le [30] ne supporte pas la notion sélective$^+$ID proposée dans [156]. Dans ce qui suit, nous allons exposer un schéma [5] compétitif à BBG et dont ces deux faiblesses n'existent pas. Nous allons relater un cryptosystème compétitive à BBG qui projette sur Z_p^* et qui supporte sélective$^+$-ID modèle. Bien entendu un schéma sécurisé dans sélective$^+$-ID modèle est aussi sécurisé dans sélective ID, l'inverse n'est pas toujours vrai. Sanitj et al [156] ont prouvé que pour convertir un schéma sécurisé dans sélective$^+$-ID modèle à un schéma sécurisé dans sélective ID cela coûte une dégradation de h (h=v-v^+, v est la longueur de l'identité cible dite du challenge, et v$^+$ est son préfixe). Qui donne plus d'avantage à l'adversaire pour attaquer le schéma en question, il est estimé selon [156] à hε.

3.2.1 Notion sélective$^+$-ID Modèle

Dans sélective$^+$-ID Modèle [156], nous donnons plus de puissance à l'adversaire, qui se produit par une modification apportée dans la phase challenge (préfixe de ID^*).

Challenge : A fait sortir deux messages égaux m_0, m_1 et une identité v+ où v+ est ID^* ou l'un de ses préfixes. Comme réponse, il reçoit le chiffrement de m sous v+, qui est choisi uniformément et indépendamment dans $\{0, 1\}$. Ce modèle est plus général que sélective ID (sID) modèle, puisque, l'adversaire autorisé à demander un challenge ciphertext non seulement pour ID^* ; mais aussi pour l'un de ses préfixes.

Un protocol sécurisé dans sélective$^+$-ID modèle est aussi sécurisé dans sélective-ID modèle. Mais l'inverse n'est pas peut être vrai.

3.2.2 Éstimation du (Decisional)k-Weak Bilinear Diffie Hellman Inversion Problem utilisé par BBG

Définition 3.2.1 *(Decisional)k-Weak Bilinear Diffie-Hellman Inversion Problem (Dk-wBDHIP*) : Soit $g \in G_2^*$. Peut on faire la séparation suivante :*

$$|Pr[g, h, g^x, g^{x^2}, ..., g^{x^k}, \hat{e}(g,h)^{\frac{1}{x}}] - Pr[g, h, g^x, g^{x^2}, ..., g^{x^k}, T]| > \varepsilon$$

Où $h, g^x, g^{x^2}, ..., g^{x^k}$ et T sont donnés.

3.2.3 Schéma sous forme de HIBE

Corps du schéma proposé

Deuxième schéma : nouveau HIBE schéma

Setup. Soient t' un niveau de sécurité convenable, (G_1, G_T) deux groupes bilinéaires.
Choisir un générateur $g \in G_1$ et soit $P_{pub_1} = g^l \in {G_1}^\star$, puis, calculer :

$$e(g,g) = x \;\; et \;\; e(g,g)^{a_1} = x^{a_1} = y_1, e(g,g)^{a_2} = x^{a_2} = y_2, ..., e(g,g)^{a_v} = x^{a_v} = y_v$$

(ou plutôt $g^{a_1}, g^{a_2}, ..., g^{a_v}$)

$$M_{pk} = \{G_1, G_T, P_{pub_1}, x, y_1, g^{a_1}, y_2, g^{a_2} ..., y_v, g^{a_v}\}, M_{sk} = \{l, a_i / 1 \leq i \leq v\}$$

L'espace du message est : $\{0,1\}^n$.
L'espace du ciphertext est : ${G_1}^* \times {G_1}^* \times \{0,1\}^n$.

Extract : Étant donné une identité $ID_A = (I_{A_1}, ..., I_{A_j}) \in {Z_p}^j$ de longueur j $\leq$ v,
d'une entité A, clé publique M_{pk}, clé maîtresse M_{sk}, puis retourner le suivant :
Pour une longueur j, calculer :

$$g^{\frac{a_1 + I_{A_1} + a_2 + I_{A_2} + ... + a_j + I_{A_j}}{l}}$$

La clé privée est :

$$(g^{\frac{a_1 + I_{A_1} + a_2 + I_{A_2} + ... + a_j + I_{A_j}}{l}}, g^{\frac{1}{l}}, g^{\frac{a_{j+1}}{l}}, ..., g^{\frac{a_v}{l}}) = (d_0, g^{\frac{1}{l}}, g^{\frac{a_{j+1}}{l}}, ..., g^{\frac{a_v}{l}})$$

Où bien elle peut être :

$$(e(g,g)^{\frac{a_1 + I_{A_1} + a_2 + I_{A_2} + ... + a_j + I_{A_j}}{l}}, e(g,g)^{\frac{1}{l}}, e(g,g)^{\frac{a_{j+1}}{l}}, ..., , e(g,g)^{\frac{a_v}{l}})$$

Pour un niveau j+1, il suffit de choisir : $s_{j+1} \in Z_p$ et calculer :

$$(g^{\frac{a_1+I_{A_1}+a_2+I_{A_2}+...+a_j+I_{A_j}+s_{j+1}(a_{j+1})+I_{A_{j+1}}}{l}}, g^{\frac{1}{l}}, g^{\frac{a_{j+2}}{l}}, ..., g^{\frac{a_v}{l}})$$

Encrypt : Étant donné m ∈ M , ID_A et M_{pk}, dérouler les étapes suivantes :

1. Choisir un random s dans Z_q.
2. Calculer :

$$e(g,g)^{s(I_{A_1}+a_1+I_{A_2}+a_2+...+I_{A_j}+a_j)} = (x^{I_{A_1}+I_{A_2}+...+I_{A_j}} y_1 y_2 ... y_j)^s$$

Le ciphertext est :

$$\mathbb{C} = (g^{ls} = {P_{pub_1}}^s, g^s, m.e(g,g)^{s(I_{A_1}+a_1+I_{A_2}+a_2+...+I_{A_j}+a_j)}) = (u', u'', v')$$

Decrypt : Étant donné $\mathbb{C}$ = (u', u",v'), ID_A, d_A, M_{pk}, suivre l'étape :

1. Calculer $e(u'', g^{(s_j-1)a_j})$ et $e(u', d_0)$ puis faire sortir :

$$m = \frac{v' e(u'', g^{(s_j-1)a_j})}{e(u', d_0)}$$

Remarques 3.2.1 :

— *Le* $(g^{\frac{a_1+I_{A_1}+a_2+I_{A_2}+...+s_j(a_j)+I_{A_j}}{l}}, g^{\frac{1}{l}}, g^{\frac{a_{j+1}}{l}}, ..., g^{\frac{a_v}{l}})$ = $(d_0, d_1, ...d_{v-1})$ *est une clé privée pour une entité en Hiérarchie (autorité bébé). Concernant un utilisateur du système, la clé privée sera* $(d_0, g^{(s_j-1)a_j})$*, si nous sommes dans un niveau j.*

— *Pour fixer t', nous pouvons suivre la même méthode déclarée dans la remarque 3.1.1.*

Exactitude

Comme nous avons :

$$\begin{aligned}
\frac{e(u'', g^{(s_j-1)a_j})}{e(u', d_0)} &= \frac{e(g^s, g^{s_j a_j}) e(g^s, g^{a_j})^{-1}}{e(g^{ls}, g^{\frac{a_1+I_{A_1}+a_2+I_{A_2}+...+s_j(a_j)+I_{A_j}}{l}})} \\
&= \frac{e(g^s, g^{s_j a_j}) e(g^s, g^{a_j})^{-1}}{e(g^s, g^{a_1+I_{A_1}+a_2+I_{A_2}+...a_{j-1}+I_{A_{j-1}}+I_{A_j}}) e(g^s, g^{s_j(a_j)})} \\
&= \frac{1}{e(g^s, g^{a_1+I_{A_1}+a_2+I_{A_2}+...a_{j-1}+I_{A_{j-1}}+a_j+I_{A_j}})}
\end{aligned}$$

Le nouveau HIBE schéma est alors correct.

3.2.4 Preuve de sécurité du nouveau HIBE schéma

La sécurité du nouveau HIBE schéma [5] est basée sur la rigidité de $Dl - BDHI_{WC}$ qui signifie Dl-BDHI avec condition, dans ce qui suit la condition est $g^{\alpha^l} = 1$. En effet :

Théorème 3.2.1 : *Supposons que (t,l,ε)-Décision $BDHI_{wc}$ s'effectue dans un groupe G_1. Alors le nouveau HIBE schéma est un (t', q_S, ε')-sélective identité, il est chosen plaintext sécurisé dans le sens de IND-sID-CPA pour des arbitraires l et q_s, sous un avantage :*

$$Adv^{nouveau\ HIBE\ schéma}(t', q_S, \varepsilon') \geq Adv^{l-DBDHI_{WC}}(t, l, \varepsilon)\ \ avec\ \ t' > t - O(lq_s\tau)$$

τ est le temps nécessaire pour réaliser l'exponentiation dans la preuve suivante :

Preuve :

Supposons que A à un avantage ε pour attaquer le nouveau HIBE schéma. Nous construisons un algorithme B qui utilise A pour résoudre le problème Décision $l - BDHI_{WC}$ dans G_1. L'algorithme B reçoit des (l+3) entrées arbitraires $(g, g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^{l-1}}, l, T) \in G_1^{*l} \times Z_q \times G_T$ telles que g^{α^l}=1, cette entré peut être obtenue soit de P_{BDHI} (avec T = e(g, $g)^{\frac{1}{\alpha}}$) ou de R_{BDHI} (avec T est uniforme et indépendante dans G_T).
Le but de l'algorithme B est de faire sortir 1 si T = e(g, $g)^{\frac{1}{\alpha}}$ et 0 sinon. L'algorithme B collabore avec A pour obtenir un gain d'une identité sélective, il travaille comme suit :

Initialisation :

Nous notons $ID^* = (I_1{}^*, ..., I_k{}^*) \in (Z_p)^k$ une identité sélective que l'algorithme A veut attaquer. Si k < v, B concatène par 1 pour obtenir exactement v qui est la longueur de l'hiérarchie.

Setup :

Comme l'algorithme A peut donner à B le $(g, g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^l}, 1 / g^{\alpha^l} = 1)$ suivant ses choix. Alors dépendamment de l'identité $ID^* = (I_1{}^*, ..., I_k{}^*)$ sélectionnée, A choisit un arbitraire j dans [1,k], par exemple avec j=2. Il calcule :

$$(g^{-I_2^*}, g^{-I_2^*\alpha}, g^{-I_2^*\alpha^2}, ..., g^{-I_2^*\alpha^l}, l/g^{\alpha^l} = 1) \tag{3.8}$$

Implicitement il calcule aussi :

$$f(\alpha) = \sum_{i=0}^{s} \alpha^i, t(\alpha) = f(\alpha) - f(0),\ \ ainsi\ \ que\ \ \frac{t(\alpha)}{\alpha} = \frac{f(\alpha) - f(0)}{\alpha} = f'(\alpha). \tag{3.9}$$

Le paramètre s doit être choisi suivant quelques requierts dans la phase 1.
Notre but est de tester si B peut faire sortir la clé privée :

$$d_A = (h^{\frac{a_1+a_2+...+a_k}{\alpha} + \frac{I_1 - I_1^* + I_2 - I_2^* + ... + I_k - I_k^*}{\alpha}}, h^{\frac{1}{\alpha}}, h^{\frac{a_{k+1}}{\alpha}}, h^{\frac{a_{k+2}}{\alpha}}, ..., h^{\frac{a_v}{\alpha}})$$
$$=(d_0, d_1, d_3, ..., d_{v-2})$$

Pour un v donné et une identité $(I_1, ..., I_v)$ (le h sera précisé après).

Premièrement B sélectionne $\gamma_1, \gamma_1, ..., \gamma_v \in {Z_p}^*$ qui doivent vérifier quelques conditions dans la phase 1

Phase 1 :

A issu au plus q_S demandes de clés privées.

Dans la première étape, il choisit une identité ID=$(I_1, ..., I_r)$, avec : $r \leq v$.

Si $r \leq k$, il sélectionne seulement r éléments de ID^* et si $r \geq k$, l'adversaire B concatène k (la longueur de I^*) par 1 comme nous avons vu précédemment.

Pour répondre à d_0, B peut suivre les étapes suivantes :

B imagine (implicitement) que chaque a_i ($1 \leq i \leq$ v) peut être écrit comme étant :

$$a_i = \gamma_i + (-1)^i \alpha^i \tag{3.10}$$

Notant que B est capable de faire ceci, puisqu'il peut choisir un convenable γ_i, tel que : $g^{\alpha^i} = g^{a_i} g^{\gamma_i}$

Nous privilégions d'utiliser $(-1)^i$ dans l'équation 3.10, car si c'est non, le $f'(\alpha) g^{\alpha^i}$ sera difficile d'être calculée .

Alors :

$$\frac{f(\alpha) - f(0)}{\alpha} \sum_{i=1}^{i=k} a_i = \frac{f(\alpha) - f(0)}{\alpha} \sum_{i=1}^{i=k} (\gamma_i + (-1)^i \alpha^i) = f'(\alpha) \sum_{i=1}^{i=k} \gamma_i + f'(\alpha) \sum_{i=1}^{i=k} (-1)^i \alpha^i \tag{3.11}$$

Premièrement $f'(\alpha) \sum_{i=1}^{i=k} \gamma_i$ peut être calculée facilement (après avoir l'exposé en g), tandis que le deuxième ne peut pas. Mais si nous la regroupons, on peut avoir alors :

$$f'(\alpha) \sum_{i=1}^{i=k} (-1)^i \alpha^i = \sum_{i=1}^{i=s} \alpha^{i-1} (-\alpha + \alpha^2 - \alpha^3 + ... \alpha^{k-1} + \alpha^k) \tag{3.12}$$

$$= -\sum_{i=1}^{i=s} \alpha^i + \sum_{i=1}^{i=s} \alpha^{i+1} - \sum_{i=1}^{i=s} \alpha^{i+2} + ... + (-1)^{k-1} \sum_{i=1}^{i=s} \alpha^{i+k-2} + (-1)^k \sum_{i=1}^{i=s} \alpha^{i+k-1} \tag{3.13}$$

Pour éliminer l'ajout des α où leur α^i est hors α^l, B doit choisir s tel que : s+k-1=l, càd s=l-k-1, ce qui implique que le grand facteur α^{i+s-1} doit être égal à 1.

Alors B peut calculer facilement :

$$g^{\frac{f(\alpha)-f(0)}{\alpha}(-I_2^*) \sum_{i=1}^{i=k} a_i} = h^{\frac{\sum_{i=1}^{i=k} a_i}{\alpha}} \tag{3.14}$$

(avec h=$g^{(\frac{f(\alpha)-f(0)}{\alpha})(-I_2^*)} = g^{f'(\alpha)(-I_2^*)}$). En exploitant le deuxième terme de 3.11 par son expression donnée dans 3.13, le 3.14 peut être calculée facilement puisqu'elle est égale à :

$$g^{(f'(\alpha)\sum_{i=1}^{i=k}\gamma_i)+(-\sum_{i=1}^{i=s}\alpha^i+\sum_{i=1}^{i=s}\alpha^{i+1}-\sum_{i=1}^{i=s}\alpha^{i+2}+...+(-1)^{k-1}\sum_{i=1}^{i=s}\alpha^{i+k-2}+(-1)^k\sum_{i=1}^{i=s}\alpha^{i+k-1})(-I_2^*)}$$
$$= g^{(f'(\alpha)\sum_{i=1}^{i=k}\gamma_i)+(-\alpha-\alpha^3-...(-1)^k\alpha^{s+k-1})(-I_2^*)}$$

En ce qui concerne la deuxième partie de d_0 qui est R = $h^{\frac{I_1-I_1^*+I_2-I_2^*+...+I_k-I_k^*}{\alpha}}$. Pour remplir cette expression il faut prendre en considération les trois exigences suivantes :

1. Le B doit être capable de faire sortir une clé exacte de ID sous laquelle tous les éléments de cette dernière figurent dans d_0.
2. Il faut que tous les I_i choisis soient différents de I_2^*.
3. il faut aussi que tout I_i (pour tout $1 \leq i \leq$ r) de l'identité demandée ID doit vérifier : $I_i \neq nI_2^*$ d'une a l'autre et ceci quelque soit n $\in$ *N*.

La raison derrière ce dernier point est le suivant : Si l'adversaire fait une recherche exhaustive pour savoir la place exacte de I_i^* pour $1 \leq i \leq v$, il a besoin de faire au minimum v recherche, ce qui est coûteux puisqu'il doit faire v ! opérations, cela est sûrement difficile surtout dans le cas d'un v grand. Alors il est conseillé de chercher un $1 \leq i \leq v$, $I_i \neq nI_2^* \ \forall$ n $\in$ *N*. Une identité qui satisfait cette condition représente le cas idéal.

Remarque 3.2.2 :

A peut choisir :

$$(g^{-I_k^*}, g^{-I_2^*\alpha^2}, g^{-I_4^*\alpha^4}, ..., g^{-I_{k-1}^*\alpha^{k-1}}, l/g^{\alpha^l} = 1)$$

Au lieu de :

$$(g^{-I_2^*}, g^{-I_2^*\alpha}, g^{-I_2^*\alpha^2}, ..., g^{-I_2^*\alpha^l}, l/g^{\alpha^l} = 1)$$

Nous avons utilisé cette dernière, càd seulement avec I_2^* pour simplifier la preuve.

Revenant maintenant au calcul de R. Pour calculer ce R, B besoin de calculer premièrement d_1. Et pour le faire, B peut calculer $\frac{f(\alpha)-f(0)}{\alpha} = f'(\alpha)$. Après il calcule :

$$g^{\frac{f(\alpha)-f(0)}{\alpha}(-I_2^*)} = g^{f'(\alpha)(-I_2^*)} = h^{\frac{1}{\alpha}} = d_1 \tag{3.15}$$

Il suffit d'exposer alors d_1 par $I_1 - I_1^* + I_2 + I_3 - I_3^* + ... + I_k - I_k^*$.

Traitons maintenant le calcul de : $d_3, d_4..., d_{v-2}$.

Comme nous avons les coefficients : $\alpha, \alpha^2, \alpha^3, ..., \alpha^{s+v-1}$, il suffit de multiplier les a_{k+1}...,a_v par $f'(\alpha)$. Effectivement, pour tout j qui dépasse l, càd l=j-x on a $\alpha^j = \alpha^x$ ou x<l, d'après le fait que α^l=1 □

Par conséquent de tous ce qu'on a vu, B peut répondre avec une clé privée sous forme :

$$d_A = (h^{\frac{a_1+a_2+...+a_k}{\alpha}+\frac{I_1-I_1^*+I_2-I_2^*+...+I_k-I_k^*}{\alpha}}, h^{\frac{1}{\alpha}}, h^{\frac{a_{k+1}}{\alpha}}, h^{\frac{a_{k+2}}{\alpha}}, ..., h^{\frac{a_v}{\alpha}}) \tag{3.16}$$

Challenge :

A fait sortir deux messages $M_0, M_1 \in G_1$. Algorithme B sélectionne un bit arbitraire b $\in$ {0,1} et un arbitraire l' $\in Z_p{}^*$. Il répond avec un ciphertext préparé comme suit :
Il calcule : $g^{(f(\alpha)-f(0))(-I_2{}^*)^s} = h^{\frac{s}{\alpha}\cdot\alpha} = h^{l'\alpha}$ = c_1, avec l'=$\frac{s}{\alpha}$
Et :

$$c_2 = MT_h^{s(a_1+a_2+...+a_k+I_1^*+I_2^*+...+I_k^*)} = T_h^{s(a_1+a_2+...+a_k+I_1^*+I_2^*+...+I_k^*)} \tag{3.17}$$

Alors si : $T_h = e(h,h)^{\frac{1}{\alpha}}$, il doit avoir :

$$e(h,h)^{\frac{s}{\alpha}(a_1+a_2+...+a_k+I_1^*+I_2^*+...+I_k^*)} = c_2 = e(h,h)^{l'(a_1+a_2+...+a_k+I_1^*+I_2^*+...+I_k^*)} \tag{3.18}$$

Le combine

$$CT = (c_1, c_2) = (h^{l'\alpha}, e(h,h)^{l'(a_1+a_2+...+a_k+I_1^*+I_2^*+...+I_k^*)}) \tag{3.19}$$

est un valide ciphertext sous ID^*.
Si T_h est uniforme dans G_1, alors CT sera indépendant du bit b.

Phase 2 :

A issu plus de demandes de clés privées (et dont il n'avait pas demandé dans la phase 1), pour un total d'au plus q_S. L'algorithme B répond comme précédemment.

Guess :

Finalement, A fait sortir un guess (estimation) b' $\in$ {0, 1}. Si b = b', alors B fait sortir 1 qui signifie que T = $e(g,g)^{\frac{1}{\alpha}}$. Sinon, il fait sortir 0 qui signifie T $\neq e(g,g)^{\frac{1}{\alpha}}$.

Si les nombres de $l+2$ entrées sont demandées de P_{BDHIP} (où T = $e(g,g)^{\frac{1}{\alpha}}$). Alors l'opinion de A est identique à son opinion dans le cas de l'attaque réelle, alors, A doit satisfaire

|Pr[b = b'] - 1/2| $> \varepsilon$. D'une autre part, si l'entrée de nombres $l+2$ sont demandés de R_{BDHIP} (où T est uniforme dans G_T), alors, Pr[b = b'] = 1/2. Par conséquent, avec g uniforme dans G_1, T est aussi uniforme dans G_T. Nous avons alors :

$$|Pr[g, g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^{l-1}}, l, \hat{e}(g,g)^{\frac{1}{\alpha}}] - Pr[g, g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^l}, l, T]| \geq |(\frac{1}{2} \pm \varepsilon) - \frac{1}{2} = \varepsilon| \quad (3.20)$$

3.2.5 Deuxième discussion

Nous débutons la discussion par le problème de Diffie-Hellman utilisé dans la preuve du théorème 3.2.1, qui est $Dl - BDHI_{WC}$. Nous avons considéré précédemment que : l >> v (v est la longueur de l'hiérarchie). Mais, cela peut rendre $Dl - BDHI_{WC}$ vulnérable à la cryptanalyse de Cheon [52] par comparaison avec $Dv - wBDHI^*$ utilisé dans le schéma BBG [30]. En effet, dans [52] Cheon a prouvé que pour les problèmes de Diffie-Hellman sous forme exponentiation, leurs complexité d'attaque à une réduction de $1/l$ (pour un l-problème sous forme exponentiation) par rapport à celle de PDL. Alors, avec $Dl - BDHI_{WC}$ où l > v (v est la longueur de l'hiérarchie), tant que l soit grand, plus qu'il peut être vulnérable à la cryptanalyse de Cheon ([52]). Pour éviter cette limite, nous proposons de prendre l=v+l', nous pouvons utiliser β où $\beta = \alpha^{l'}$ au lieu de α dans les demandes du problème (g^{α^i}), ceci peut réduire $Dl - BDHI_{WC}$ à $Dv - BDHI_{WC}$, on peut même faire une réduction encore plus fine.

Les relations entre les problèmes utilisées dans les schémas BB1, BBG et le nouveau HIBE schéma peuvent se voir comme suit :

$$l - BHIP \ \ (BB1) \longrightarrow^1 l - wBDHI^* \ \ (BBG) \longrightarrow^2 l - BDHI_{WC} \ \ (nouveau \ \ HIBE \ \ schéma) \quad (3.21)$$

La relation 1 a été prouvée dans [23], tandis que 2 est facile d'être justifiée. Par conséquent, de 3.21 on a : $Dl - BHIP \longrightarrow^1 Dl - wBDHI^* \longrightarrow^2 Dl - BDHI_{WC}$.

Même si le schéma BBG [30], est basé sur un problème de Diffie Hellman qui est rigide par comparaison avec le nouveau HIBE schéma (cela se réfère peut être à l'utilisation d'un couplage asymétrique) ; mais BBG à deux faiblesses : la première réside au fait qu'il projette sur Z_p^* au lieu de Z_p, ce qui limite la sélection de l'identité cible d'être le challenge, comme on ne peut nul part utiliser le bit 0. En plus, le BBG ne supporte pas $s^+ID - CPA$; par contre le nouveau HIBE schéma [5] est comme BB1 supporte cette notion. Selon ce qui est annoncé dans [156], pour rendre BBG $s^+ID-CPA$, les auteurs lui ont apporté une simple modification. La preuve de cette dernière a crée une dégradation dans la sécurité suivant un facteur v, où v est le nombre maximal du niveau utilisé par un HIBE. Pour ne pas obtenir cette dégradation,

les auteurs ajoutent v-k facteurs ou plutôt $(v-k)Exp_{G_1}$ dans l'original schéma (v est la longueur maximale de l'hiérarchie, tandis que k est la longueur de l'identité sélectionnée ID^*).
Par contre avec notre proposition (nouveau HIBE schéma), on n'a pas besoin de cela, puisque notre schéma est $s^+ID-CPA$. En plus que cela, notre schéma offre une compétition à BBG [30], pour le savoir, nous contentons de compter dans ce qui suit la complexité de BB1, BBG, et le nouveau HIBE schéma :

	Extract	Encrypt	Decrypt
BB1	$(2k+3)Exp_{G_1}$	$(2k+1)Exp_{G_1}+1Exp_{G_T}$	(k+1)coup + $kMul_{G_T}$
BBG	$3Exp_{G_1}$	$(k+2)Exp_{G_1}$ + $1Exp_{G_T}$	2coup
Nouveau HIBE schéma	$2Exp_{G_1}$ ou $2Exp_{G_T}$	$(k+2)Exp_{G_T}$ + $2Exp_{G_1}$	2 coup +$1Mul_{G_T}$ + $1Exp_{G_1}$

TABLE 3.4 – Calcul de complexité des schémas : BB1, BBG et le nouveau HIBE schéma dans le niveau k

Remarques 3.2.3

— *Dans ce tableau, nous n'avons pas tenu en compte quelques complexités (division des couplages, multiplication par $y_1y_2...y_k$ dans notre schéma, multiplication par g_3 dans BBG...)*

— *Les notations utilisées dan ce tableau sont les mêmes que celles utilisées dans la partie 3.1.5.*

Selon ce tableau, le nouveau HIBE schéma est plus efficace que BB1 et même que BBG. Or, Exp_{G_T} qu'on le compte comme $Exp_{Z_{p^{k'}}}$ (càd dans le corps fini) est plus réduit que Exp_{G_1} (càd dans la courbe elliptique).
L'efficacité est visible dans l'Extract et l'Encrypt (par rapport aux deux schémas BB1 et BBG). Pour le Decrypt, nous dépassons un petit peu BBG ; mais du fait de ce qu'on a dit en haut (point de vue sécurité), nous pouvons dire que le nouveau HIBE schéma est aussi plus efficace que BBG.

3.3 Quatrième schéma sous le modèle Random Oracle pour l'IBE

Jusqu'à nos jours, les crypto systèmes sous le modèle Random Oracle sont au nombre de trois : Boneh et Franklin [22], Sakai et Kasahara [154], Boneh et Boyen (full version : version complète [25] où H(ID) est un Random Oracle). Le maillon faible de la notion modèle Random [125], nécessite de réduire dès que possible les faiblesses des crypto systèmes présentés sous ce modèle. Il en ressort que le schéma de Boneh et Franklin ne peut pas être dissocié de la projection sur une courbe elliptique, ce qui limite l'utilisation de cette dernière et ceci réagit directement sur la sécurité. Pas mal d'études didactiques visant à contourner ce problème

ont été proposées. Par exemple, Michel Scott [159] propose d'utiliser $H_1(ID) = cH_0(ID)$, avec H_0 est le haché d'un point ponctuel au lieu d'un point d'une courbe elliptique, le c est un cofacteur. Malheureusement, cette étude est seulement valable pour le couplage de Tate au lieu de celui de Weil qui est originalement utilisée par Boneh et Franklin. En se basant sur [106], le [41] est considéré comme une autre solution. Cette étude consiste de hacher sur les courbes ordinaires au lieu des courbes supersingulières. Ce résultat paraît comme important, mais elle est loin d'être certes et complète. En outre, le [41] suggère que la preuve de sécurité reste encore valable si nous intégrons le modèle Random Oracle, ce qui demande une étude approfondie. Ainsi, le [41] fonctionne seulement avec la caractéristique 3 et donc il présente une restriction.

En second lieu, le schéma de Sakai et Kasahara est un crypto système qui ne maintient pas la faiblesse précédente puisqu'il projette sur Z_p. Toutefois, il est basé dans les études de simulation sur un problème mineur : k-BDHIP qui a comme complexité d'attaque $O(q/k)$ (PDL à O(q)) suite aux résultats de Cheon[52]. Ce qui rend sa sécurité fragile contre un attaqueur malin et même passif.

Dans la partie suivante, nous allons proposer [6] un quatrième schéma pour l'IBE sous le modèle Random Oracle. Le nouveau schéma ne projette pas sur une courbe elliptique, ainsi il utilise un problème plus rigide que k-BDHIP.

3.3.1 Corps du schéma proposé

Quatrième schéma de l'IBE sous le modèle Random Oracle

Setup. Étant donné un paramètre de sécurité k,

les paramètres générateurs (publiques) se calculent comme suit :

— Générer : quatre groupes cycliques G_1, G_2 et G_T d'ordres premier q.

Deux isomorphismes ψ_2, ψ_3 respectivement de G_2 à G_1 et de G_3 à G_1.

Enfin un couplage bilinéaire

ê : $G_1 \times G_2 \longrightarrow G_T$.

Sélectionner un générateur arbitraire $g_1 \in {G_1}^\star$ et soit g_2= $\psi_2(g_1)$, g_3= $\psi_3(g_1)$

— Sélectionner un nombre arbitraire s qui est le résidu quadratique de a càd a=s^2,

après calculer Pub_1 = ${g_1}^s$, Pub_2 = ${g_1}^a$

— Calculer quatre fonctions de Hachage : $H_1 : \{0,1\}^* \longrightarrow Z_q^*$, $H_2 : G_T \longrightarrow \{0,1\}^n$,

$H_3 : \{0,1\}^n \times \{0,1\}^n \longrightarrow Z_q^*$ et $H_4 : \{0,1\}^n \longrightarrow \{0,1\}^n$ pour un entier n > 0.

L'espace du message est : M = $\{0,1\}^n$.

L'espace du ciphertext est C= $G_1^* \times G_1^* \times \{0,1\}^n \times \{0,1\}^n$.

La clé publique est :

$$M_{pk} = \{q, G_1, G_2, G_T, \psi_2, \psi_3, \hat{e}, n, g_1, Pub_1, Pub_2, \hat{e}(g_1, g_3) = l, \hat{e}(g_1, g_3)^a = l^a, H_1, H_2, H_3, H_4\}.$$

La clé maîtresse est : M_{sk} = {s, a}.

Extract : Étant donné $ID_A \in \{0,1\}^*$ d'une entité A, M_{pk} et M_{sk}, l'algorithme sélectionne un arbitraire r_{ID_A} et il fait sortir

$$d_A = (r_{ID_A}, (g_2{}^{-r_{ID}} g_3)^{\frac{H_1(ID_A)}{s} + \frac{s}{H_1(ID_A)}}) = (r_{ID_A}, d_{ID_A})$$

Encrypt : Étant donné un plaintext m ∈ M, ID_A et M_{pk}, suivre les étapes suivantes.

1. Choisir un arbitraire $\sigma \in \{0,1\}^n$ et un r = $H_3(\sigma, m)$.
2. Calculer g_1^{sr}, g_2^r = $\psi_2(g_1)^r$
3. Le ciphertext est

$$C = (g_1^{sr} = Pub_1{}^r, g_2^r, \sigma \oplus H_2(l^{r(a+(H_1{}^2(ID_A))}) = \sigma \oplus H_2(l^{ra} l^{rH_1{}^2(ID_A)}), m \oplus H_4(\sigma))$$
$$= (u,v,x,y)$$

Decrypt : Étant donné un ciphertext C = (u,v,x,y), ID_A, d_A et M_{pk}, suivre les étapes :

1. Calculer

$$z = \hat{e}(u^{H_1(ID_A)}, d_{ID_A}) \hat{e}(Pub_2{}^{r_{ID_A}} g_1{}^{r_{ID_A} H_1{}^2(ID_A)}, v)$$
$$= \hat{e}(u^{H_1(ID_A)}, d_{ID_A}) \hat{e}(g_1{}^{ar_{ID_A}} g_1{}^{r_{ID_A} H_1{}^2(ID_A)}, \psi_2(g_1)^r)$$

2. Calculer $x \oplus H_2(z) = \sigma'$.
3. Calculer $y \oplus H_4(\sigma') = m'$ et r'=$H_3(\sigma', m')$
4. Vérifier si u $\neq g_1^{sr'}$ ou v$\neq g_2^{r'}$, faire sortir $\perp$, sinon retourner m' comme étant le plaintext.

Exactitude :

Comme nous avons :

$$\hat{e}(u^{H_1(ID_A)}, d_{ID_A}) \hat{e}(g_1{}^{ar_{ID_A}} g_1{}^{r_{ID_A} H_1{}^2(ID_A)}, g_2^r)$$
$$= \hat{e}(g_1{}^r, (g_2{}^{-r_{ID_A}} g_3)^{(H_1{}^2(ID_A)+a)}) \hat{e}(g_1{}^{(H_1{}^2(ID_A)+a) r_{ID_A}}, g_2^r)$$
$$= \hat{e}(g_1{}^r, g_2{}^{-r_{ID_A}(H_1{}^2(ID_A)+a)}) \hat{e}(g_1{}^r, g_3^{H_1{}^2(ID_A)+a}) \hat{e}(g_1{}^{(H_1{}^2(ID_A)+a) r_{ID_A}}, g_2^r)$$

Le quatrième schéma de l'IBE est alors correct.

3.3.2 Preuve de sécurité

La sécurité du quatrième schéma de l'IBE [6] est résumé dans la rigidité du problème 4-BDHEP. La réduction est similaire à la preuve de Boneh et Franklin [22] et celle de Sakai-Kasahara [154] (où sa sécurité a été prouvée dans [49]), comme avec [49] nous tenons en compte dans notre preuve la remarque ou plutôt la révision de Galindo [86] (voir le chapitre 1 pour le rappel).
Le théorème suivant représente le degré de sécurité du quatrième schéma de l'IBE :

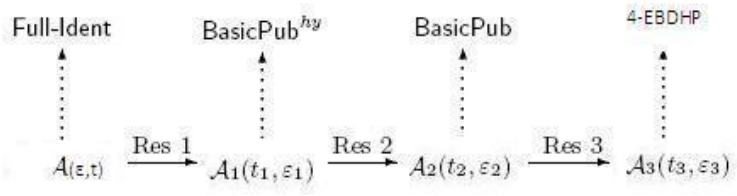
Théorème 3.3.1 : *Le quatrième schéma de l'IBE est sécurisé contre un IND-ID-CCA adversaire avec le fait que $H_i(1 \leq i \leq 4)$ sont des random oracles et 4-EBDHP est asymptotique rigide. Supposons qu'il existe un IND-ID-CCA adversaire A contre le quatrième schéma de l'IBE et qui a un avantage ε(k) dans un temps t(k). Supposons aussi que pendant l'attaque, A fait au moins q_d demandes du déchiffrement et au plus q_i demandes de H_i pour $(1 \leq i \leq 4)$ respectivement (le H_i est peut être demandé directement par A ou indirectement par : la méthode d'extraction des demandes, demandes du déchiffrements ou par une opération du challenge). Alors il existe un algorithme A_3 qui peut résoudre le problème 4-EBDHP avec un avantage $Adv_{A_3}(k)$ et dans un temps $t_{A_3}(k)$, où :*

$$Adv_{A_3}(k) = \frac{1}{q_2(q_3+q_4)}[(\frac{\varepsilon(k)}{q_1^2}+1)(1-\frac{2}{q})^{q_d}-1]$$
$$t_{A_3}(k) \leq t(k) + O((q_3+q_4)(n+logq) + q_d(\tau_1+\tau_2+\chi))$$

Avec : q est l'ordre de G_1 et n est la longeur de σ. Le q_i, i ∈ {1, 2, 3} et τ_i où i ∈ {1, 2}, χ seront déterminés selon les lemmes 3.3.1, 3.3.2 et 3.3.3.

Preuve :

Comme avec Boneh et Franklin [22], dans notre preuve nous suivons immédiatement les trois réductions suivantes :



Pour distinguer ces Res_i i ∈ {1,2,3} nous combinons trois lemmes :
Dans le lemme 3.3.1 nous prouvons que s'il existe un IND-ID-CCA adversaire, qui est capable d'attaquer le quatrième schéma de l'IBE en lançant un adaptatif chosen ciphertext attaque comme il a été définie en [22] et dans la partie 1.6.1, alors il existe un IND-CCA adversaire

qui peut casser le schéma **BasicPub**^{hy}. Le but de cette étape est de prouver que l'extraction des demandes de la clé privée n'aide pas l'adversaire.

Dans le lemme 3.3.2, nous prouvons que si le type IND-CCA d'adversaires existe, alors il doit exister un IND-CPA adversaire qui correspond à **BasicPub** schéma en lançant le chosen plaintext attaque. Nous présenterons le but de cette étape dans la fin du lemme 3.3.3. Nous notons que l'adversaire ne peut pas bénéficier de l'extraction des clés privées.

En fin, dans le lemme 3.3.3, nous prouvons que si **BasicPub** schéma est non sécurisé contre le IND-CPA adversaire, alors le correspond asymptotic 4-EBDHP peut, peut être, facilement attaquer.

Lemme 3.3.1 : *Supposons que H_1 est un Random Oracle et qu'il existe un IND-ID-CCA adversaire A contre le quatrième schéma de l'IBE qui a l'avantage $\varepsilon(k)$ et qui peut faire au plus q_1 demandes différentes de H_1 (noté que H_1 peut être demandé directement par A ou indirectement par un extraction des demandes du déchiffrement ou encore d'après l'opération challenge). Alors il existe un IND-CCA adversaire A_1 qui se déroule dans un temps $O(A) + q_d(\tau_1 + \chi)$ contre* ***BasicPub***^{hy} *schéma qui a un avantage au moins $\frac{\varepsilon(k)}{{q_1}^2}$.*

Le **BasicPub**^{hy} se caractérise par les trois algorithmes : **keygen**, **encrypt** et **decrypt**.

keygen ou **setup :**

Étant donné le paramètre de sécurité qui est k, le challenger donne les paramètres suivants à A_1.

$$K_{pub} =< q, G_1, G_2, G_T, \psi_2, \psi_3, k_2 = ord(\psi_2), k_3 = ord(\psi_3), \hat{e}, n, h_i$$

$$pour\ \ i \in \{0, ...q_1\}, g_1, g_2, g_3, Pub_1 = g_1^s, Pub_2 = g_1^{s^2}, \hat{e}(g_1, g_3)^a, \hat{e}(g_1, g_3), H_2, H_3, H_4 >$$

encrypt :

Sélectionner arbitrairement $\sigma \in \{0,1\}^n$ et calculer : r = $H_3(\sigma, m)$, ${g_1}^{sr}$ et ${g_2}^r$.

Le ciphertext est :

$$C_i = ({g_1}^{sr}, {g_2}^r, \sigma \oplus H_2((l)^{r(a+({h_i}^2(ID_A)))}) = \sigma \oplus H_2(l^{ra}{l^{rh_i}}^2(ID_A)), m \oplus H_4(\sigma))$$

$$= (u, v, x_i, y)$$

decrypt :

Étant donné le ciphertext $C_i = (u, v, x_i, y), ID_A, d_A$ et M_{pk}, suivre les étapes :

1. Calculer z_i=ê$(u^{h_i(ID_A)}, d_{ID_A})$ê$(P_{ub2}{}^{r_{ID_A}} {g_1}^{r_{ID_A}{h_i}^2(ID_A)}, {g_2}^r = v)$
2. Calculer $x_i \oplus H_2(z_i) = \sigma_i'$.
3. Calculer $y \oplus H_4(\sigma_i') = m'$ et $r_i' = H_3(\sigma_i', m')$
4. Vérifier si u $\neq {g_1}^{sr_i'}$ ou v $\neq {g_2}^{r_i'}$, faire sortir $\perp$, sinon retourner m' comme étant le plaintext.

Nous construisons un adversaire A_1 qui lance un IND-CCA attaque dans le $BasicPub^{hy}$ schéma, en utilisant la clé publique K_{pub} et l'aide de A. L'attaque est comme suit :
Premièrement, nous notons par **BasicPu**$b_{1-H_1}{}^{hy}$ quand l'adversaire A_1 à un avantage pour demander au challenger, les demandes de son $H_1(ID_{A_1})$. Si le challenger répond à lui, sa réponse est comme suit :

$$(r_{ID_{A_1}}, (g_2{}^{-r_{ID_{A_1}}} g_3)^{\frac{H_1(ID_{A_1})}{s} + \frac{s}{H_1(ID_{A_1})}}, H_1(ID_{A_1})) \tag{3.22}$$

L'algorithme A_1 peut calculer : $(g_2{}^{-r_{ID_{A_1}}} g_3)^{\frac{s}{H_1(ID_{A_1})}}$ puisqu'il connaît Pub_2, Pub_1, ψ_2 et ψ_3.
Après, il peut calculer aussi :

$$(g_2{}^{-r_{ID_{A_1}}} g_3)^{\frac{H_1(ID_{A_1})}{s} + \frac{s}{H_1(ID_{A_1})}} . (g_2{}^{-r_{ID_{A_1}}} g_3)^{-\frac{s}{H_1(ID_{A_1})}}$$

Alors l'algorithme A_1 peut calculer facilement : $(g_2{}^{-r_{ID_{A_1}}} g_3)^{\frac{1}{s}}$
Pour un h_i, soit a_{h_i} tel que $r_{h_i} + a_{h_i}$=$r_{ID_{A_1}}$, nous avons donc : $(g_2{}^{-r_{h_i} - a_{h_i}} g_3)^{\frac{1}{s}}$.
A_1 peut calculer une clé exacte si : k_2=$\frac{a_{h_i}}{s}$ ou a_{h_i} = ord(g_2), sinon il peut échouer.
A_1 peut vérifier cela, d'après la formule : $\psi_2(Pub_1)^{k_2} = \psi_2(g_1)^{a_{h_i}}$
Nous notons que la seconde partie de la clé : $(g_2{}^{r_{h_i}} g_3)^{\frac{s}{h_i}}$ est peut être calculée facilement si nous avons $Pub_1, Pub_2, \psi_2, \psi_3$.
L'algorithme A_1 simule l'étape **Setup** du quatrième schéma de l'IBE à A en lui fournissant la clé publique :

$$M_{pk} = \{q, G_1, G_2, G_T, \psi_2, \psi_3, \hat{e}, n, g_1, g_2, g_3, Pub_1, Pub_2, \hat{e}(g_1, g_3)^a, e(g_1, g_3), H_1, H_2, H_3, H_4\}$$

Où H_1 est un random oracle contrôlé par A_1. A_1 ne connaît pas la clé maîtresse {s,a}.
L'adversaire A fournit des demandes à H_1 a n'importe quel moment. Les demandes seront calculées d'après l'algorithme suivant :
H_1-query (ID_i) :

A_1 a une liste de (ID_i, h_i, d_i) indexée par ID_i comme nous avons expliqué précédemment. Nous référons cette liste par $H_1{}^{list}$. Cette liste est initialement vide. Quand A demande H_1 dans un point ID_i, A_1 répond comme suit :

1. Si ID_i apparaît dans $H_1{}^{list}$ sous la forme (ID_i, h_i, d_i), alors A_1 répond par : $H_1(ID_i) = h_i$.
2. Sinon, A_1 sélectionne un entier arbitraire h_i(i > 0) de M_{pk} qui n'a pas été choisi, il utilise la méthode précitée et il stocke le triple dans la liste. Après, A_1 répond par $H_1(ID_i) = h_i$.

Phase 1 :

Dans cette étape A demande soit : une extraction de clé ou des demandes du déchiffrement (decryption càd sans clé privée). A_1 répond à ses demandes comme suit :

Extraction de clés pour des (ID_i) : Premièrement, A_1 examine la liste ${H_1}^{list}$. Si ID_i n'est pas dans la liste, alors A_1 demande $H_1(ID_i)$. Après, A_1 examine la valeur d_i. Si $d_i \neq \perp$, A_1 répond par d_i ; sinon, A_1 perd le jeu (**Événement 1**).

Demande de Decryption pour des (ID_i, c_i) : Premièrement, A_1 examine la liste ${H_1}^{list}$. Si ID_i ne figure pas dans la liste, A_1 demande $H_1(ID_i)$. Si $d_i = \perp$, alors A_1 demande le déchiffrement de $c_i = (u, v, x_i, y)$ au challenger et simplement retient le plaintext donné par le challenger à A directement. Sinon, A_1 déchiffre le ciphertext.

Challenge :

Dans n'importe quel temps, A décide de finir la Phase 1, il sélectionne ID_{ch} et deux messages (m_0, m_1) (dont il veut les présenter comme étant challenge), qui ont des longueurs égales. En se basant sur les demandes de H_1, A_1 répond comme suit :

1. Si les demandes de H_1 sont issues et d_{ch}=$\perp$, A_1 continue,
   — Sinon, A_1 perd le gain (**Événement 2**).
2. Si le uplet correspond à ID_{ch} est dans la liste ${H_1}^{list}$ (et donc $d_{ch} \neq \perp$), alors A_1 perd le gain (**Événement 3**).

A_1 donne au challenger le couple (m_0, m_1) comme un message cible (challenge). Le challenger, choisit arbitrairement b $\in \{0,1\}$, il chiffre m_b et il répond avec le ciphertext C_{ch} = (u', v', x',y'). Alors A_1 donne C_{ch} à A.

Phase 2 :

A_1 continu de répondre aux demandes, en utilisant la même méthode que la Phase 1. Nous notons que l'adversaire ne peut pas demander l'extraction de ID_{ch} (pour lequel $d_{ch} \neq \perp$) et le déchiffrement des demandes de (ID_{ch}, C_{ch}).

Observation 3.3.1 : *A_1 ne peut pas perdre le gain dans la phase 2, comme il n'a pas le droit de répondre aux demandes de ID_{ch} et ceux de C_{ch}.*

Guess :

A donne une estimation b' pour b. A_1 fait sortir b' comme étant son estimation.

Cette simulation (étude) est identique à une attaque réelle si elle n'a pas échouée quelque part.

Affirmation 3.1 :

Si l'algorithme A_1 n'a pas échoué pendant la simulation, alors l'opinion de l'algorithme A est identique à son opinion dans l'attaque réelle.

Preuve : Les réponses de A_1 à H_1 sont uniformes et distribuées d'une manière indépendante dans Z_q, elles sont comme dans l'attaque réelle puisque les réponses de A_1 sont arbitraires et valides, cela si A_1 n'a pas échoué.

Il nous reste alors le calcul des probabilités non échouées pendant la simulation.

$$Pr[A_1\ n'a\ pas\ échoué] = Pr[\neg\ événement_1 \wedge \neg\ événement_2 \wedge \neg\ événement_3]$$
$$= Pr[\neg\ événement_1].Pr[\neg\ événement_1/(\neg\ événement_2 \wedge \neg\ événement_3)]$$
$$= \frac{1}{{q_1}^2}$$

Qui se réalise pendant un temps $t_1 = O(A) + q_d(\tau_1 + \chi)$

Où τ_1, χ sont respectivement le temps du calcul de l'exponentiation et du couplage.

Le lemme suivant est le fruit des résultats de Fujisaki et Okamoto (Théorème 14 dans [81]). Avec le fait que $BasicPub^{hy}$ se résulte après avoir appliqué la transformation de Fujisaki-Okamoto à la version basique du schéma 3.3.1 (sans effectuer H_3, H_4 dans sa version complète). Nous donnons ci-après la version basique du schéma 3.3.1.

Lemme 3.3.2 : *Soient H_3, H_4 deux Random Oracles. Soit A_1 un IND-CCA adversaire contre $BasicPub^{hy}$ qui a un avantage ε_1(k) comme il a été défini dans le Lemme 3.3.1. Pendant le temps $t_1(k)$, supposons que A_1 fait : au plus q_d demandes du déchiffrement, et q_3 et q_4 demandes pour H_3 et H_4 respectivement. Alors il existe un IND-CPA adversaire A_2 contre le schéma BasicPub (du quatrième schéma de l'IBE). On défini ce dernier par les trois algorithmes :* ***keygen***, ***encrypt*** *et* ***decrypt****.*

keygen :

Étant donné le paramètre de sécurité k.

1. L'étape de préparation est la même que $BasicPub^{hy}$, mais cette fois nous éliminons H_3 et H_4. Par contre il faut laisser H_2.
2. Calculer une fonction de hachage $H_2 : G_T \longrightarrow \{o,1\}^n$, M = $\{0,1\}^n$. L'espace du ciphertext est : $G_1^* \times \{0,1\}^n \times \{0,1\}^n$. La clé publique est :
   $M_{pk} = \{q, G_1, G_2, G_T, \psi_2, \psi_3, \hat{e}, n, g_1, g_2, g_3, Pub_1, Pub_2, \hat{e}(g_1, g_3) = l, \hat{e}(g_1, g_3)^a = l^a, h_i\ \ ou\ \ i \in \{o,1\}^n, H_2\}$

   La clé secrète (ou plutôt la clé maîtresse) est M_{sk} = {s,a}

encrypt

Choisir un arbitraire r $\in Z_q$ et calculer g_1^{sr} et g_2^r.

Le ciphertext est :

$$C_i = (g_1^{sr}, g_2^r, m \oplus H_2(l)^{r(a+({h_i}^2(ID_A)))}) = (u, v, x_i)$$

decrypt :

Étant donné le ciphertext $C_i = (u, v, x_i)$, ID_A, d_A et M_{pk}, suivre les étapes :

1. Calculer :

$$z_i = \hat{e}(u^{h_i(ID_A)}, d_{ID})\hat{e}(Pub_2{}^{r_{ID}} {g_1}^{r_{ID}{h_i}^2(ID_A)}, v = {g_2}^r) = e(g_1, g_2)^{a+h_i(ID_A)^2}$$

2. Calculer : $x_i \oplus H_2(z_i) = m$.

Selon [81] A_2 à l'avantage suivant : $\varepsilon_2(k)$ et un temps t_2 tels que :

$$\varepsilon_2(k) \geq \frac{1}{2(q_3+q_4)}[(\varepsilon_1(k)+1)(1-\frac{2}{q})^{q_d} - 1]$$

$$t_2(k) \leq t_1(k) + O((q_3+q_4).(n+logq))$$

Lemma 3.3.3 : *Supposons qu'il existe un IND-CPA adversaire A_2 contre le BasicPub qui a un avantage $\varepsilon_2(k)$, comme il a été défini dans le Lemme 3.3.2, des demandes aux plus, q_2 fois H_2 (où H_2 est un Random Oracle). Alors il existe un algorithme A_3 qui peut résoudre le problème 4-BDHE avec un avantage d'au moins $\frac{2\varepsilon_2}{q_2}$, qui se déroule dans un temps $O(time(A_2) + q_d\tau_2)$.*
Où τ_2 est le temps qui permet de calculer l'exponentiation dans G_2.

L'algorithme A_3 donne comme entrée l'arbitraire 4-BDHE instance :

$$\{q, G_1, G_2, G_T, \psi_2, \psi_3, \hat{e}, k_2, k_3, Q_1, Q_2, P_{pub} = {Q_1}^x, Q_3, /k_3 - k_2 = ord(g_1)\}$$

Où x est un élément arbitraire de Z_q. Nous déterminerons Q_1, Q_2, Q_3 après.
La clé privée est :

$$d_{partiel} = (r_{ID}, (Q_2^{-r_{ID}} Q_3)^x)$$

Nous donnons à l'algorithme A_3 : { $g_1, g_2, g_3, {g_1}^x, {g_1}^{x^2}, {g_1}^{x^3}$ }, l'algorithme A_3 trouve $\hat{e}(g_1, g_2)^{\frac{1}{x}}$ et $\hat{e}(g_1, g_3)^{\frac{1}{x}}$ (nous notons que nous pouvons calculer $\hat{e}(g_1, g_2)^{\frac{1}{x}}$ et $\hat{e}(g_1, g_3)^{\frac{1}{x}}$, d'après ψ_2, ψ_3) en communiquant avec A_2 comme suit :

L'algorithme A_3 calcule f(x)=$\sum^2{}_{i=0} c_i x^i$ avec :

$$c_0 = \begin{cases} 0 & A_3 \text{ utilise l'aide de } A_2 \text{ (qui utilise de son tour l'aide de } A_1\text{),} \\ & \text{mais, } A_1 \text{ ne reçoit pas } d_{ID_{A_3}} \text{ pour son choix de } H_1 \text{ dans la phase 1 (lemme 3.3.1).} \\ 1 & A_1 \text{ reçoit } d_{ID_{A_3}} \text{ pour } H_1 \text{ (en utilisant la même méthode de transitivité).} \end{cases}$$

Nous supposons que A_1 reste sous la disposition de A_2 dès que A_3 ne termine pas son travail
Si c_0 = 0, A_1 peut premièrement calculer les demandes dans la phase 1 en utilisant la manière déclarée ci-après, à condition que le challenger publie :

$$M_{pub} = \{q, G_1, G_2, G_T, \psi_2, \psi_3, \hat{e}, k_2, k_3, Q_1 = {g_1}^x, P_{pub} = {Q_1}^x, Q_3, /k_3 - k_2 = ord(g_1)\}$$

Alors nous avons, pour chaque h_j, $\frac{f(x-h_j)-f(-h_j)}{x} = c_1 - 2c_2h_j + c_2x$=E
Nous avons aussi :

$$f(x - h_j) - f(-h_j) = x(c_1 - 2c_2h_j) + c_2x^2 = F$$

Nous remarquons que ${g_1}^E$ et ${g_1}^F$ peuvent être calculés facilement.
Par suite :

$${g_1}^{k_3E} = ({g_1}^{k_3(c_1x+c_2x^2)}{g_1}^{-k_2(2c_2xh_j)})^{\frac{1}{x}} = ({g_3}^{(c_1x)}{g_2}^{-(2c_2x)h_j})^{\frac{1}{x}}{g_3}^{(c_2x)}$$

Alors si nous posons $Q_2 = {g_2}^x$ et $Q_3 = {g_3}^{c_1x}$.

$$({g_1}^{k_3E})^{h_j}({g_3}^{c_2x})^{-h_j} = ({Q_2}^{-2c_2h_j}Q_3)^{\frac{h_j}{x}} = ({Q_2}^{-r_{h_j}}Q_3)^{\frac{h_j}{x}};\ avec\ r_{h_j} = 2c_2h_j$$

En utilisant la même méthode, nous avons aussi :

$$({g_1}^{k_3F})^{\frac{1}{h_j}}({g_3}^{c_2x^3})^{-\frac{1}{h_j}} = ({Q_2}^{-r_{h_i}}Q_3)^{\frac{x}{h_j}}$$

Comme résultat, nous trouvons :

$$({g_1}^{k_3E})^{h_i}({g_3}^{c_2x})^{-h_i}({g_1}^{k_3F})^{\frac{1}{h_j}}({g_3}^{c_2x^3})^{-\frac{1}{h_j}} = ({Q_2}^{-r_{h_j}}Q_3)^{\frac{h_j}{x}+\frac{x}{h_j}}$$

Alors on peut répondre aux demandes et on peut calculer trivialement :

$$\hat{e}({P_{pub}}^{h_j}, ({Q_2}^{-r_{h_j}}Q_3)^{\frac{h_j}{x}+\frac{x}{h_j}}) = \hat{e}(Q_1, {Q_2}^{-r_{ID}}Q_3)^{a+{h_j}^2}$$

Mais si c'est non, càd si $c_0 \neq 0$, on a alors :
Phase 1 se déroule suivant le lemme 3.3.1
L'adversaire A_3 peut calculer :

$$({Q_2}^{-r_{h_j}}Q_3)^{\frac{h_j}{x}+\frac{x}{h_j}}$$

en utilisant ce qui a été cité au dessus, en plus :

$$f(x - h_j) - f(-h_j) = c_1 x + c_2 x^2 - 2c_2 h_j$$

Dans les deux cas ($c_0 \neq 0$ et $c_0 = 0$), l'algorithme A_3 peut calculer : $(Q_2{}^{-r_{h_j}} Q_3)^{\frac{h_j}{x} + \frac{x}{h_j}}$

Alors nous avons :

$$\hat{e}(P_{pub}{}^{h_j}, (Q_2{}^{-r_{h_j}} Q_3)^{\frac{h_j}{x} + \frac{x}{h_j}}) = \hat{e}(Q_1, Q_2{}^{-r_{ID}} Q_3)^{a + {h_j}^2}$$

Et

$$\hat{e}(P_{pub}{}^{h_j}, d_{partiel}{}^{\frac{1}{h_j}} d_{complète}{}^{h_j}) = \hat{e}(Q_1, Q_2{}^{-r_{ID}} Q_3)^{a + {h_j}^2}$$

Avec : $d_{complete}$= $(g_1{}^{k_3 E})(g_3{}^{c_2 x})$= $(Q_2{}^{-r_{h_j}} Q_3)^x$ qui peut être calculé aisément par A_3.

Récapitulons, M_{pub} est une valide clé publique pour BasicPub.

Maintenant A_3 commence à répondre aux demandes en utilisant l'algorithme suivant :

$H_2 - query(X_i)$: A n'importe quel moment l'adversaire A_2 demande des demandes du random oracle H_2. Pour répondre à ses demandes, A_3 à une liste de résultats nommée ${H_2}^{list}$. Chaque entrée dans la liste est un candidat sous la forme(X_i, ζ_i) indexé par X_i. Pour répondre aux demandes des X_i, A_3 fait les opérations suivantes :

1. Si dans la liste il y a un condidat indexé par X_i, alors A_3 répond avec ζ_i
2. Sinon, A_3 choisit arbitrairement des entiers $\zeta_i \in \{0,1\}^n$ et insère un nouveau candidat (X_i, ζ_i) dans la liste. Il répond à A_2 avec ζ_i.

Challenge :

L'algorithme A_2 fait sortir deux messages (m_0, m_1) de longueurs égales dont lesquelles il veut les présenter comme étant un challenge. A_3 choisit un entier arbitraire R $\in \{0,1\}^n$ et un élément arbitraire r $\in Z_q$ et il définie C_{ch} = $({Q_1}^r, {Pub_1}^r, R)$. A_3 donne C_{ch} comme un challenge à A_2. Nous observons que le déchifrement de C_{ch} est :

$$R(H_2(\hat{e}({Pub_1}^r, Q_2{}^{-r_{ID}} Q_3)^x)^{-1} \hat{e}({Pub_1}^r, Q_2{}^{-r_{ID}} Q_3)^{\frac{1}{x}}))$$

Guess :

Après, l'algorithme A_2 fait sortir une estimation, A_3 sélectionne un arbitraire (X_i, ζ_i) à partir de H_{list}. Rappelons que :

$$\hat{e}(Pub_1, d_{partial}) = \hat{e}(Pub_1, (Q_2{}^{-r_{ID}} Q_3)^x) = \hat{e}(g_1, g_2)^{-r_{ID} x^4} . \hat{e}(g_1, g_3)^{x^4}$$

Et

$$\hat{e}(Q_1, (Q_2{}^{-r_{ID}} Q_3)^x) = \hat{e}(g_1, g_2)^{-r_{ID} x^3} . \hat{e}(g_1, g_3)^{x^3}$$

Les deux ne peuvent être calculé que si on réussi 4-BDHEP

Mais les calculs :

$$\hat{e}(Pub_1, d_{complète}) = \hat{e}(Pub_1, (Q_2{}^{-r_{ID}} Q_3)^{\frac{1}{x}}) = \hat{e}(g_1, g_2)^{-r_{ID}x^2}.\hat{e}(g_1, g_3)^{x^2}$$

Et

$$\hat{e}(Q_1, d_{complète}) = \hat{e}(Q_1, (Q_2{}^{-r_{ID}} Q_3)^{\frac{1}{x}}) = \hat{e}(g_1, g_2)^{-r_{ID}x}.\hat{e}(g_1, g_3)^{x}$$

sont faciles à calculer.

Soit *D* l'événement sous lequel l'algorithme A_2 fait sortir des demandes pour H_2(ê($Pub_1, d_{partial}$)) pour n'importe quel point dans la simulation précédente. Pour tester, si cet événement semble à l'attaque réelle, nous avons besoin de deux affirmations suivantes [22] :

Affirmation 3.2 : *Pr[D] dans la simulation précédente est égale à Pr[D] dans l'attaque réelle.*

Affirmation 3.3 : *Dans l'attaque réelle nous avons Pr[D]* $\geq 2\varepsilon_2(k)$.

Alors comme réclamation, on peut dire que A_3 *produit une réponse correcte s'il a réussi le calcul de 4-BDHEP et s'il travaille sous une attaque réelle. Cette dernière à une probabilité de reussite d'au moins* $\frac{2\varepsilon_2(k)}{q_2}$.

Le temps pour réaliser le lemme 3.3.3 est $O(time(A_2) + q_d\tau_2)$ où τ_2 est le temps pour calculer l'exponentiation

Comme résultat de tout ce qu'on a vu :

Preuve du ***lemme 3.3.1*** + *Preuve du* ***lemme 3.3.2*** + *Preuve du* ***lemme 3.3.3*** = *Preuve du* ***théorème 3.3.1***.

3.3.3 Étude de perfomance

L'ojectif premier de cette partie est de comparer le quatrième schéma de l'IBE [6] avec les anciens qui sont sous le modèle Random Oracle. Nous avons pour ambition dans le cadre de la proposition du quatrième schéma de l'IBE de contourner les faiblesses dont les anciens schémas sous le modèle Random Oracle souffrent, nous nous concentrerons alors sur la sécurité.

Comparaison au niveau sécurité

Pour faire la comparaison au niveau sécurité, nous sélectionnons comme barème :

1. Type du problème bilinéaire de Diffie Hellman

2. Projection ou non sur la courbe elliptique

3. Symétrique ou Asymétrique du couplage

Nous avons choisit de nous baser sur ces points pour les raisons suivantes :
Étudier la rigidité (selon le type) des problèmes de Diffie-Hellman utilisés dans les études de simulation de ces crypto systèmes, nous donnent leurs poids contre un adversaire passif (CPA) et un adversaire malin (CCA2).
Ainsi, la projection sur une courbe elliptique limite la sélection des courbes elliptiques utilisées, ce qui pose un problème de sécurité, puisque, trouver un $H(ID)^3 + aH(ID) + b$ sous forme d'un résidu quadratique est très difficile sauf pour certains cas très particuliers.
En ce qui concerne le type des couplages, l'attaque MOV [134] à un danger non souhaitable. Les courbes supersingulières sont produites par une utilisation des couplages symétriques, elles sont alors vulnérables à cette attaque, par contre, les courbes ordinaires conduisent aux couplages asymétriques. En plus, il est prouvé dans [83] que les couplages asymétriques sont plus désirables que ceux des couplages symétriques.

Schémas/points du barème	**Boneh et Franklin**	**Sakai et Kasahara**
Problème bilinéaire de Diffie Hellman	BDHP	q-BDHIP
Projection sur la courbe elliptique	oui	non
Symétrique/Asymétrique du couplage	Non, version asymétrique avec [86]	Non, version asymétrique avec [49]
Schémas/points du barème	**BB1**	**Quatrième schéma** de l'IBE [6]
Problème bilinéaire de Diffie et Hellman	BDHP (non sûr)	4-EBDHP
Projection dans la courbe elliptique	non	non
Symétrique/Asymétrique du couplage	Asymétrique	Asymétrique

TABLE 3.5 – Comparaison au niveau sécurité entre les quatres schémas de l'IBE sous le modèle des Random Oracle

Rappelons que [22] utilise le couplage symétrique ; mais Galindo pendant sa révision [86] a utilisé le couplage asymétrique. Chen et Cheng [49] lors de leurs preuve de sécurité du schéma de Sakai et Kasahara utilisent aussi le couplage asymétrique.

Coup d'oeil sur la comparaison

Pour comparer les trois schémas : BB1 [23], Sakai Kasahara (version [49]), Boneh et Franklin [22] et le quatrième schéma de l'IBE au niveau du point 2, nous nous sommes basé sur les réductions suivantes :

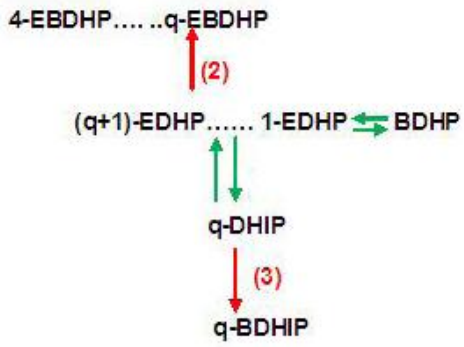


FIGURE 3.1 – Relations entre quelques problèmes de Diffie-Hellman.

Le : k-A $\longrightarrow$ k-B veut dire que si k-A est résolu dans un temps polynomial, alors k-B l'est aussi.

Les relations en vert sont prouvées dans [49].

Les relations en rouge sont triviales, en effet, démontrer par exemple la relation (3) est d'après : Étant donné $(g, g^x, g^{x^2}, g^{x^3}, ..., g^{x^k})$, si on peut calculer $g^{\frac{1}{x}}$ on peut calculer aussi ê$(g, g)^{\frac{1}{x}}$.

Nous notons que (2) est peut être fait de la même manière, il faut seulement faire attention à la réduction suivante : (q+1)-EDHP $\longrightarrow$ q-EDHP

Mais suivant les résultats de Cheon [52], travaillé avec un q grand n'est pas bon, puisque le q-EDHP (ainsi que d'autres problèmes sous forme exponentiation càd ceux qui utilisent $g^{x^{k_i}}$) à une complexité d'attaque O(1/q) par rapport à PDL, il va réduire plus vite. En effet, tant que q est grand plus que q-EDHP soit fragile que PDL.

Dans [49] le q est relié à q_H (puisque l'oracle est construit dans le BasicPub à partir de q-BDHIP [49]). Suivant [118] nous avons besoin de q_H=2^{50} pour un niveau de sécurité égale à celui de 80. Et il faut l'augmenter pour les niveaux de sécurité plus élevés. Alors puisque nous avons : $2^2 < 2^{50}$ et donc le 4-EBDHP garantie plus de sécurité que q-BDHIP. Nous pouvons dire la même chose pour BDHP, comme il a le même niveau de sécurité que 1-EDHIP (voir la figure 3.1).

Tenons en compte cela, on peut noter que seuls les schémas : quatrième schéma de l'IBE, Boneh et Franklin et BB1 sont efficaces. Mais, le schéma de Boneh et Franklin ne peut pas être séparé de la projection sur la courbe elliptique. Cela limite la sélection de cette dernière, il reste alors que notre schéma et le BB1 [23] (dans [26] Boyen a présenté les bienfaits de ce schéma par comparaison avec [22][154]) qui soient efficaces.

Mais, on n'est pas sûr du problème de Diffie Hellman (ou plutôt des paramètres) utilisé par BB1, puisqu'on n'a pas une preuve exacte (la preuve a été faite seulement avec sID-CPA).
Dans l'attente de cela, nous comparons notre schéma avec BB1 (version CPA et sous le modèle sélective-ID [23]).

Comparaison avec BB1 selon le point de vue complexité

	BB1 (version IBE) [23]
Params	$2Exp_G + 1Mul_{Z_p} + 1coup + 1Ex_{F_{q^k}}$
Extract	$3Mul_{Z_p} + 2Exp_G$
Encrypt	$1Exp_{F_{p^k}} + 1Mul_{Z_p} + 3Exp_G$
Decrypt	$2coup + 1inv_{F_{p^k}} + 1Exp_{F_{q^k}} + 1Exp_G$
Somme	$3coup + 1div_{F_{p^k}} + 3Exp_{F_{p^k}} + 8Exp_G + 5Mul_{Z_p}$

TABLE 3.6 – Complexité de BB1

	Quatrième schéma de l'IBE [6]
Params	$2Exp_G + 1Mul_{Z_p} + 1coup + 1Ex_{F_{q^k}}$
Extract	$2div_{Z_p} + 2Exp_G$
Encrypt	$2Exp_G + 2Exp_{F_{p^k}} + 2Mul_{Z_p}$
Decrypt	$2Mul_{Z_p} + 4Exp_G + 2coup$
Somme	$3coup + 2div_{Z_p} + 2Exp_{F_{p^k}} + 10Exp_G + 5Mul_{Z_p}$

TABLE 3.7 – Complexité du quatrième schéma de l'IBE

Dans les tableaux 3.6 et 3.7 on a utilisé les mêmes notation que la partie 3.1.5

Analyse des résultats

Dans l'opération $Exp_{F_{p^k}}$, l'exposant est dans F_p. Mais sa base est dans F_{p^k}. Aussi, pour $div_{F_{p^k}}$ nous faisons la division dans F_{p^k}. Le F_{p^k} est un corps fini construit en utilisant le quotient $F_p[X]/P(X)F_p[X]$).
Suivant le tableau 3.6, on peut équilibrer entre BB1 et le quatrième schéma de l'IBE, puisque :

Somme(BB1)-Somme(quatrième schéma de l'IBE) = $1div_{F_{p^k}} + Exp_{F_{p^k}} - 2div_{Z_p} - 2Exp_G$

On peut équilibrer entre $1div_{F_{p^k}}$ et $2div_{Z_p}$ puisque dans $1div_{F_{p^k}}$ nous faisons la division de deux polynômes, après nous la réduirons suivant le module P(x) (qui est une division cachée). □
Nous avons un extra $2Exp_G$ par comparaison avec BB1, du fait de l'utilisation de r_{ID} dans l'Extract, dans le but, de nous aider dans la preuve CCA2. Si nous éliminons ce paramètre, nous éliminons alors le $1Exp_G$ dans l'Extract et $2Exp_G$ dans le Decrypt qui est en somme de $3Exp_G$. Par contre, le BB1 a été prouvé seulement CPA dans le modèle sélective ID (introduit par [45]), cette dernière est une notion faible [87], pour prouver que BB1 est CCA2 sous le modèle random oracle il faut réviser sa preuve. □

3.4 Conlusion

Nous venons de présenter des schémas efficaces avec couplage pour le chiffrement basé sur l'identité. Notre innovation en ce sens se résume en trois schémas.

— Le premier est sous le modèle sélective ID et l'approche Commutative Blinding ; mais plus réduit que BB1 et même que BB2, qui sont les seuls tracés dans la littérature sous ce modèle.

— Le deuxième schéma est un HIBE sous le modèle sélective ID, nous avons montré dans la partie 3.2.5, l'avantage au niveau complexité que présente notre schéma par rapport à BBG. En plus, nous avons contourné quelques faiblesses dont souffre ce dernier, notre schéma projete sur Z_p, au lieu de Z_p^*, ainsi, il supporte la propriété sélective ID^+.

— Le troisième schéma proposé est sous le modèle Random Oracle, il est compté comme un quatrième schéma pour l'IBE sous ce modèle. Ce schéma projecte sur Z_p contrairement à Boneh et Franklin, ainsi il est basé sur un problème plus rigide que celui utilisé par Sakai Kasahara. Par conséquent, le nouveau schéma est le plus sécurisé et il offre une compétition au niveau complexité au standard BB1 [25] (version CPA).

Chapitre 4

Protection contre le Key Escrow et les attaques par canaux cachés

Ce chapitre a pour objet de prévenir quelques défaillances de la cryptographie basée sur l'identité (IBC), en particulier l'IBE. En effet, ce dernier souffre de :

— Problème de Key Escrow

— Peu d'études pour la sécurité physique

— Stockage susceptible de la clé maîtresse

— Problème de révocation des clés

— Non résistance aux ordinateurs quantiques

Dans ce chapitre nous allons focaliser nos travaux sur le premier et le deuxième points.
Dans les premières parties du chapitre on récapitule brièvement nos propositions pour résoudre le problème de Key Escrow. Pourtant, la littérature a connu deux propositions fondamentales, l'idée d'Al-Riyami et al [2] et celle de Gentry [92]. Malheureusement, les deux présentent des carences de ce qu'on appelle Daniel of Decryption (DoD) qui permet à l'autorité de remplacer les clés générées en défendant le déchiffrement du message. D'où l'exploitation d'un travail fait par Joseph K. Liu et al [129] qui reste encore non satisfaisant. En se basant sur le simple protocole de Diffie Hellman, on propose dans la deuxième partie, une méthode qui gère le Key Escrow et qui résiste aussi bien au DoD (Daniel of Decryption).
La réalisation pratique de l'IBE dans certains appareils électroniques tels que les cartes à puce, devient un but délicat auprès des attaques physiques.
Dans la troisième partie du chapitre on rappele le fonctionnement des attaques auxiliaires.
Dans la quatrième partie on introduit nos idées concernant la sécurité de l'IBE contre les attaques physiques, on commence par citer une méthode pour réussir l'attaque DPA sur les couplages, puis une autre sur la cryptographie IBC, on examine par la suite les contre-

mesures dédiées par la littérature pour défendre les attaques DPA appliquées au couplage en proposant des alternatives convenables.
La dernière partie du chapitre est réservée à une étude appropriée pour défendre les attaques DFA appliquées au couplage et cela quelque soit le degré de plongement pair.

4.1 Peu d'état de l'art sur le Key Escrow

4.1.1 Généralité

Intégrer une tierce personne pour authentifier les utilisateurs dans les schémas à clé publique PKC, pose la question sur sa sûreté et sa confiance. Dans la PKI, le problème ne se pose pas, puisque l'intégrité de la clé publique est garantie d'après l'authentification suivant un certificat numérique qui contient la clé et les informations liées, publiées suivant un annuaire publique. Pour l'IBE, le problème se pose avec acuité. En outre, la clé privée générée est définie comme résultat d'une opération mathématique qui relie l'identité de l'utilisateur (celle du récepteur) et la clé maîtresse (master key) enfantée par l'autorité PKG. Le principal inconvénient de cette méthode est le manque de confiance rendue à la PKG. C'est encore pire que la PKC traditionnelle, depuis la clé secrète de chaque utilisateur est générée par la PKG, elle peut envahir l'identité d'un utilisateur, ou déchiffrer n'importe quel texte chiffré.

4.1.2 Key Escrow en brève

Le Key Escrow se résume, au fait qu'une entité qu'on a chargé avec toutes confiance, nous aide pendant toute nos communications (authentifier, réserver ou distribuer nos clés soit discrètement ou publiquement), est capable de lire nos messages, si elle connaît nos clés secrètes, c'est le cas de l'IBE original. Ou bien, remplacer nos clés publiques afin de déchiffrer nos messages, c'est le cas de la PKC traditionelle.

Ses niveaux

Il y a trois niveaux de Key Escrow :

— ***Niveau I :*** *L'autorité peut facilement calculer la clé secrète des utilisateurs, elle peut illégalement prendre leur place sans être détectée. Dans ce cas le Key Escrow ne peut être résolue.*

— ***Niveau II :*** *L'autorité ne sait pas, ou bien ne peut pas calculer la clé privée des utilisateurs. Mais elle peut les envahir, en générant des fausses clés publiques, aussi, sans être détectée.*

— ***Niveau III :*** *L'autorité ne sait pas ou bien ne peut pas calculer la clé privée des utilisateurs. Mais elle peut les envahir dans les preuves de sécurité, en leur générant des fausses clés publiques.*

Dans le sens de cette dernière, deux types d'attaques peuvent être générées :

Type-1 : *Des adversaires internes dans le système (excepte le récepteur légitime) peuvent accèder à la clé maîtresse, mais ils leur interdit de choisir aucune clé publique pour l'utiliser dans le challenge du ciphertext.*

Type-2 : *Un honnête-mais-curieux PKG peut accéder à la clé maîtresse, il peut enregistrer la clé publique pendant le challenge.*

Pour remédier au problème de Key Escrow, dans la PKC traditionnelle, par exemple Shamir et après Micali ont proposé l'utilisation d'un Key Escrow partiel, et donc il faut que l'autorité fasse des recherches exhaustives pour accèder à la clé complète. Sous le cadre IBE, parmi les solutions proposées, celles de distribuer la clé maîtresse sur plusieurs PKG, mais ceci exige une forte infrastructure. Ainsi, peut être qu'il y aura une coopération entre ces entités. Depuis 2003, d'autres solutions sont connues ; mais nous nous contentons de citer hâtivement dans ce chapitre les trois proposition : Certificateless Public Key Cryptography (CL-PKC) [2], Certificate Based Encryption(CBE) [92], Self-Generated-Certificate Public Key Cryptography (SGC-PKC) [129].

4.1.3 Anciennes méthodes pour résoudre le Key Escrow et leurs faiblesses

Certificateless Public Key Cryptography (CL-PKC) [2]

Elle est premièrement introduite par Al-Riyami et Paterson en ASIACRYPT 2003 [2], ils ont relié l'IBE (en général IBC) avec la méthode traditionnelle PKC en conservant l'avantage de non utilisation de Certificat. En CL-PKC on génère en général trois clés, une clé partielle par la PKG, deux autres par l'utilisateur (récepteur), ces dernières sont : clé publique générée à partir de la clé partielle afin de l'utiliser comme témoin puisqu'on n'utilise pas de certificat, on la publie pour tout le monde comme en PKC, la deuxième clé est secrète et conservée chez l'utilisateur. Ce principe est à peu près similaire à celui du self certificat introduit par Girault en 1991 [95]. (Plus de détails se trouvent dans [2])

Pendant la même année 2003 et lors d'EUROCRYPT, Gentry a proposé une autre solution [92] :

Certificate Based Encryption(CBE)

Cette méthode est plus proche au contexte du modèle traditionnel PKC, puisqu'elle utilise aussi une autorité qui fournit une certification (signature) mais implicitement (sans certificat). Toutefois, dans ce modèle on chiffre notre message par deux clés : la clé de la PKG et l'autre de l'utilisateur (clé publique du récepteur) et on envoie le message chiffré avec une certification (signature) afin de garantir l'origine de la clé publique. La signature est faite seulement par une partie de la clé secrète, alors que le reste de la clé reste chez l'utilisateur (récepteur). Pour plus d'informations voir [92].

Malheureusement ces deux méthodes sont classées dans les niveaux II et III puisqu'elles nécessitent la génération d'une clé publique de l'utilisateur sans certificat. Donc si cette dernière est publiée dans le Params, la PKG peut tricher dans sa clé privée (pour les deux méthodes) puisqu'il n'existe aucune vérification. En plus, ce cas demande au PKG d'être on-line et de contacter le récepteur avant le chiffrement, il est alors non efficace. Si encore c'est le récepteur qui donne la clé publique, cette dernière peut tomber facilement à l'attaque du Man in the Middle (l'homme du milieu).
Dans les deux cas, les deux méthodes souffrent de ce qu'on appelle : Denial-of-Decryption (DoD) attaque. Car en plus des raisons citées, une active attaquante Eve (hors les membres de la PKG) peut se présenter comme étant le récepteur Bob. Alors la destinataire Alice qui n'est pas au courant de ceci, continue à envoyer les messages chiffrer à Bob, tandis que ce dernier ne peut pas les déchiffrer. Par conséquent, même si Eve ne peut pas accèder à ces messages puisqu'elle n'a pas l'identité de Bob, mais elle fait rater l'occasion à Bob de lire ses messages. Donc qu'est ce qu'on peut faire pour résoudre le Key Escrow dans l'IBE, en résistant à l'attaque DoD ? la réponse est dans ce qui suit :

Self-Generated-Certificate Public Key Cryptography (SGC-PKC) [129]

Elle est proposée en 2007 [129] par Joseph K. Liu, Man Ho Au et Willy Susilo. D'une manière similaire à CL-PKC et PKC, la destinataire Alice calcule un certificat implicite et auto-signé à partir de deux clés : une partielle de la PKG, psk=$(psk^{(1)}, psk^{(2)})$ et l'autre publique de Bob (le récepteur) qui est pk = (g^x, g_1^x) = $(pk^{(1)}, pk^{(2)})$ et elle vérifie en même temps le certificat autosigné en calculant : $e(pk^{(1)}, g^1) =^? e(pk^{(2)}, g)$ et $e(V, g) =^? e(g_2, pk^{(2)})\ e(U, R_\pi)\ e(m' \prod_{i \in M}^{m_i}, R_m)$, tous ces paramètres sont extraits du [129]. Donc si ce certificat est invalide alors Alice demande encore à Bob de lui envoyer sa clé publique. Mais rien n'est garanti puisque la clé publique de Bob n'est pas certifiée par une PKI.

Dans un soucis de complétude, nous rappelons ici brièvement quelques notions qu'on pourra utiliser pour exploiter notre proposition qui sera citée dans la section 4.2.

4.1.4 Notions supplémentaires

Fonction à sens unique

Une fonction f est dite à sens unique si, étant donné un x dans un intervalle de recherche, il est facile de calculer f(x), mais l'inverse n'est pas possible. Autrement, étant donné un g = f(x) pour un élément x aléatoire, il est difficile de trouver x tel que f(x) = g. Une fonction à sens unique f est dite à trappe si, à l'aide d'un secret qui s'appelle dans ce cas trappe, il est possible de calculer un antécédent, appelé préimage de tout élément dans l'image de f.
Dans toute la section 4.2 nous noterons par H une fonction à trappe.

Protocole de Diffie-Hellman

Le protocole de Diffie Hellman est résumé dans le schéma suivant :

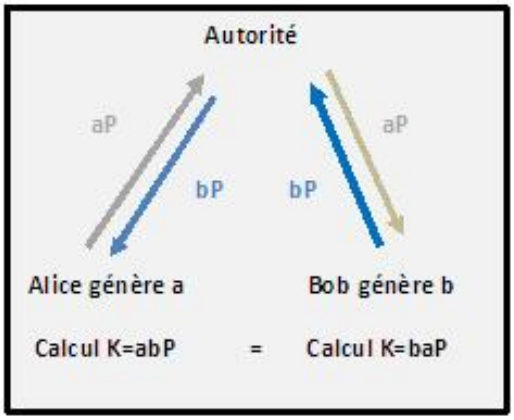


FIGURE 4.1 – Protocole de Diffie Hellman sous l'utilisation d'une autorité.

À partir du paramètre publique P (ou (g,p)) et d'un secret a connu d'elle seule, Alice calcule aP (ou g^a mod p) et l'envoie à Bob en utilisant l'aurotité. Bob fait de même à partir de son secret b calcule bP (ou g^b mod p) et l'envoie à Alice (par l'intermédiaire de l'autorité). Alice et Bob peuvent ainsi crèer une clé commune K = abP = baP (ou $K = (g^a(modp))^b = (g^b(modp))^a$).

Schéma de Yinxia Sun, Futai Zhang, et Joonsang Baek : sans couplage [166]

Schéma de Yinxia Sun et al[166]

Setup : msk = s $\in Z_q$, mpk = y = $g^s \in Z_p^*$, (q|(p-1), alors Z_p^* est d'ordre q),
$pub = \{g, p, q, H_0(), H_1(), H_2(,), H_3()\ \}$.

KeyDer(s,id) :

1. Choisir τ_0, τ_1 dans Z_q, calculer $w_0 = g^{\tau_0}$, $w_1 = g^{\tau_1}$

2. Calculer $d_0 = \tau_0 + sH_0$(id $||$ w_0) et $d_1 = \tau_1 + sH_1$(id $||$ w_0 $||$ w_1)
3. Faire sortir D_{id} = $< d_0, P_{id} >$ ou P_{id} = $< w_0, w_1, d_1 >$.

   (Pid doit être publique, comme clé partielle)

SetUK' :

sk = $<$ x $\in Z_q$, $d_0 >$, pk = u = $< g^x, w_0, w_1, d_1 >$.

Enc(M, id , pk) :

1. Vérifier si g^{d_1} = $w_1 y^{H_1(id||w_0||w_1)}$, sinon, $\perp$ est retourner.
2. Sélectionner $\rho \in Z_q$, calculer r = H_3(ρ ||M ||u||id) $\in Z_q$.
3. Faire sortir C = $< C_1, C_2 >$ = $< g^r$, (M|| ρ) $\oplus$ $H_2(w_0^r$ y$^{H_0(id||w_0)r}$, $u^r)>$.

Dec($< C_1, C_2 >$, $< x, d_0 >$) :

1. Recouvrir $Z_1 = C_1^{d_0}$ et $Z_2 = C_1^x$, retirer (M'|| ρ') = $C_2 \oplus H_2(Z_1, Z_2)$.
2. Retourner M' si $C_1 = g^{H_3(\rho'||M'||u||id)}$, $\perp$ sinon.

4.2 Notre méthodologie pour éviter le Key Escrow et le DoD

Contrairement, à ce qu'on avait vu précédement, concernant l'utilisation d'un certificat implicite, dans notre méthode [7] nous n'allons pas utiliser celle-ci, nous nous baserons seulement sur le protocole de Diffie Hellman. Nous proposons d'échanger deux clés aP et bP par l'intermédiaire de la PKG, afin d'être encadré par cette dernière, et pour arrêter les attaques plus vastes comme avec PKC (c'est à dire que n'importe quelle personne à l'avantage de remplacer nos clés). Aprés être rassurés de nos clés (d'aprés la méthode de la partie 4.2.2), on peut passer nos messages en intégrant la clé commune qu'on calcule suivant le protocole de Diffie Hellman dans les cryptosystèmes de l'IBE. Cela évite bien le Key Escrow et il résiste à DoD, puisque la clé résultante (commune) est secrète, en plus, son intégrité est vérifiée. Nous résumons notre procédure [7] dans les trois méthodes qui suivent.

4.2.1 Méthode d'envoi des deux clés

Avant d'envoyer aP, Alice l'enregistre chez une entité confidente. Alice lui donne un paramètre privé $r_{ID_{Alice}}$. L'entité stocke quelque part $r_{ID_{Alice}}$ muni de l'identité d'Alice qui peut être son adresse email, son numéro de téléphone, peut importe, l'identité enregistrée est celle qui apparaîtra dans la communication Alice-Bob.
Supposons maintenant qu'Alice à l'identité de Bob, et veut lui envoyer la clé aP, elle contacte tout d'abord un PKG (consulte son site) afin d'obtenir les paramètres publics.

Remarque 4.2.1 *: On appelle la première entité utilisée par Alice entité d'enregistrement, tandis que la deuxième où elle contacte pour extraire les paramètres publics est une PKG connu usuellement dans l'IBE. C'est mieux que la première soit différente de la deuxième.*

Dans un premièr temps, Alice extrait les paramétres $\ll K_q, G, H_1, P \gg$ où : K est le corps choisi, q un paramétre de sécurité choisi en vérifiant certaines conditions de sécurité (suivant par exemple le NIST), G un groupe d'ordre q, P un générateur de G, et à la fin H_1 est une fonction à trappe.
Alice choisit un a $\in K_q^*$ secret, elle calcule le aP, mais puisque cette clé n'est pas certifiée par une PKI pour indiquer qu'elle est bien envoyée par Alice, et donc avant de l'envoyer à Bob suivant le protocole de Diffie Hellman. Alice choisit deux nombres secrets s_{Alice} et s_{Bob}, le premier est réservé à toutes les communications d'Alice, tandis que le deuxième concerne seulement celle d'Alice-Bob.
Alice calcule :

$$< H_1(s_{Bob} + ID_{Bob}), H_1(s_{Bob})P, s_{Alice}r^{-1}_{ID_{Alice}}P, s_{Alice}P, (s_{Alice} + s_{Bob} + a)P > \quad (4.1)$$

Qui l'envoie à Bob en précisant l'entité à contacter pour que Bob puisse vérifier l'émetteur. À la réception, Bob contacte l'entité précisée, celle d'enregistrement (on l'appelle première entité) afin de vérifier le transmetteur Alice, il l'envoi

$$< s_{Alice}r^{-1}_{ID_{Alice}}P, s_{Alice}P > \quad (4.2)$$

Si $r_{ID_{Alice}}.s_{Alice}r^{-1}_{ID_{Alice}}$P=$s_{Alice}$P, l'entité envoie oui à Bob, sinon, elle envoi $\perp$.
Lorsque Bob vérifie le transmetteur Alice, il contacte la PKG (on l'appelle deuxième entité), Bob ne peut plus retirer aP sauf s'il est le vrai Bob, la PKG vérifie son identité en calculant

$$H_1(H_1^{-1}(H_1(s_{Bob} + ID_{Bob})) - ID_{Bob})P = H_1(s_{Bob})P \quad (4.3)$$

Après elle extrait s_{Bob}P, elle le donne à Bob, ce dernier calcul en toute confiance

$$(s_{Alice} + s_{Bob} + a)P - s_{Bob}P - s_{Alice}P = aP \quad (4.4)$$

Le s_{Bob} ne peut pas être retiré qu'après avoir inverser H_1, ce qui est irréalisable, seulement la PKG est capable de le faire.

De la même manière Bob envoie sa clé publique bP à Alice.

Maintenant qu'on est sûr que s'il existe un changement, seulement l'autorité PKG (deuxième, la première entité est seulement une vérificatrice) qui est capable de le faire. Puisqu'elle est la seule qui peut inverser le H_1 et extraire le s_{Bob}, et donc elle sera la seule qui peut tricher les utilisateurs. Alors avant d'integrer la clé commune qu'on calcule d'après le protocole de Diffie Hellman dans les cryptosystèmes d'IBE pour chiffrer et déchiffrer nos messages, il faut tout d'abord s'assurer que nos clés ne sont pas remplacées par la PKG.

4.2.2 Méthode de vérification

Aprés avoir bien reçu les deux clés aP et bP par les deux entitées, il leur sera simple de calculer abP = baP. Mais il faut vérifier que ces deux clés ne sont pas falsifiées. Pour le faire, par exemple dans le cas d'Alice, elle choisit un nombre aléatoire m dans K_q^* et secret, après, elle envoie à Bob :

$$< r_{ID_{Alice}}^{-1}(mbaP + s_{Bob}P), mbaP + s_{Bob}P > \tag{4.5}$$

Le $r_{ID_{Alice}}^{-1}$ a pour but d'authentifier Alice en contactant l'autorité choisie par elle (première entité). Bob qui a déjà reçu s_{Bob}P, extrait facilement mbaP, et il calcule : mbaP-aP.
Quatre cas à envisager :

Cas de changement

1. b^{-1}(mbaP)-aP = (maP)b'b^{-1}-aP (si b est changé par b')
2. b^{-1}(mbaP)-aP = (maP)bb^{-1}-a'P = (ma-a')P (si a est changé par exemple par a')
3. b^{-1}(mbaP)-aP = (maP)b'b^{-1}-a'P (si a et b sont changés respectivement par a' et b')

Cas du non changement

4. b^{-1}(mbaP)-aP = (maP)bb^{-1}-aP = $(m-1)aP$ (or ni a ni b sont changés)

Bob envoie $< r_{ID_{Bob}}^{-1}(b^{-1}$(mbaP)-aP + $s_{Alice}P)$, b^{-1}(mbaP)-aP + $s_{Alice}P >$ à Alice et donc c'est seulement en quatrième cas qu'Alice peut s'assurer que a et b n'ont pas été changés. Puisque Alice peut facilement extraire (m-1)aP et le comparer avec b^{-1}(mbaP)-aP. Si les vérifications sont invalides, Alice et Bob contactent la PKG comme entité responsable du changement, ils peuvent ainsi régler les choses avec elle.

4.2.3 Méthode de chiffrement et de déchiffrement

Maintenant qu'ils ont vérifié le non changement des clés publiques aP et bP, Alice et Bob calculent K=abP = baP. Alors la clé secrète d'Alice et Bob est bien K (connu seulement par Alice et Bob), et on peut l'intégrer dans n'importe quel cryptosystème d'IBE.

• Première utilisation : Cryptosystème avec couplage, sans Key Escrow, résiste à DoD

Par exemple, intégrons cette clé dans le schéma basique de Boneh et Franklin [22], on aura :
Pour chiffrer un message M $\in \{0,1\}^n$, choisir un nombre r $\in Z_q$. Le message est chiffré comme suit :

$$\text{C} = < rP, M \bigoplus K \bigoplus H_2(g^r) > = < U, V >, \quad \text{avec} \quad \text{g} = e(Q_{ID}, P_{pub}) \in G_2^*$$

Pour déchiffrer ce message en utilisant les paramétres publics :

$< q, G_2, G_2, e, n, P, P_{pub} = sP, Q_{ID}, H_2 >$ et une clé privée $d_{ID} = sQ_{ID} \in G_1^*$

Calculer :

$$\text{V} \bigoplus H_2(e(d_{ID},\text{U})) = M \bigoplus K$$

En utilisant la clé K, l'extraction du message est d'après :

$$M \bigoplus K \bigoplus K = \text{M}.$$

Nous notons que :

$$\text{e}(d_{ID},\text{U}) = \text{e}(sQ_{ID},\text{rP}) = e(Q_{ID}, P)^{sr} = e(Q_{ID}, P_{pub})^r = g_{ID}^r.$$

Nous comparons dans le tableau suivant, notre approche avec les méthodes citées dans ce chapitre.

	Implicite Certificat	Pas de Key Escrow	Résiste à DOD	Nécessite la confiance de l'autorité
PKI	Non	Oui	Oui	Oui
IBE	Oui	Non	Oui	Oui
PKC	Oui	Non	Non	pas utiliser
CL PKC	Oui	Oui	Non	Non
CBE	Oui	Oui	Non	Non
SGC PKC	Oui	Oui	pas garanti	Non
Notre	Non	Oui	Oui	Oui/Non

TABLE 4.1 – Comparaison entre notre approche et les méthodes existantes

Dans SGC PKC le DOD n'est pas garantie, puisque, si nous générons un $\perp$ (symbole qui indique l'échec), le [129] demande de contacter directement Bob ce qui n'est pas garantie puisque Bob n'est pas certifié.
Dans notre approche on a mis Oui/Non, le Oui pour la première autorité (vérifieuse) et le Non pour la deuxième autorité (PKG usuel).

Suite au profit de la flexibilité apportée par la clé générée par notre méthode précitée, qui est utile pour diverses utilisations comme nous pouvons les voir ci-après. Nous faisons trois expériences en utilisant différents schémas, la première est celle qu'on a juste présenté, nous l'avons nommé première utilisation. Dans cette utilisation, le cryptosystème utilisé fonctionne avec les couplages, bien entendu, l'intégration de la clé générée suivant la section 4.2 dans ce cryptosystème, nous permet d'éviter le Key Escrow ainsi que de défendre à DoD.
Ici, nous étudions la performance de cette utilisation en présentant deux autres expériences, ainsi que leurs efficacité.

4.2.4 Étude de performance

Première comparaison

Dans [117] Aniket Kate et Ian Goldberg ont présenté une méthode qui permet de distribuer la clé maîtresse entre plusieurs autorités. Dans un premier temps, cette idée a pour but de conserver la clé maîtresse, dans le deuxième temps, elle permet d'obtenir des crypto systèmes sans Key Escrow puisque la clé privée cible ne peut pas être connue qu'après un calcul mathématique des clés partielles générées à partir des morceaux de la clé maîtresse. Donc, avec l'hypothèse de l'existence des autorités honnêtes dans le système, ce calcul sera inaccessible aux adversaires externes ainsi qu'internes. Le tableau suivant expose la comparaison entre la première utilisation où on a appliqué le schéma de Boneh Franklin et le [117] qui est aussi tracé avec le même schéma.

	Schéma [117]	Schéma de la première utilisation
Multiplication	t	12
Multiplication Scalaire	5(t+1)+5	11
Inversion	t	3
Couplage	4 (moins des cas) (t+4) (pire des cas)	2

TABLE 4.2 – Comparaison entre le schéma [117] et celui de Boneh et Franklin (version basique) muni de la clé résultante

Analyse des résultas

Tout d'abord, t est le nombre de noeux utilisés. Il est clair que pour t $\geq$ 12, notre schéma est le plus efficace.

Prenons maintenant le cas où t = 3, il est le nombre des noeux le plus bas qu'on peut choisir. En utilisant les résultats de [109], nous avons :

$$nP = (n-1)ECDBL + \frac{(n-1)}{3}ECADD = (n-1)(7Mu+5Sq) + \frac{(n-1)}{3}(12Mu+2Sq). \tag{4.6}$$

Pour un n par exemple égal aussi à 3 (cas le plus bas), on aura :

	Schéma [117]	Schéma de la première utilisation
Multiplication	3+25(14+$\frac{2}{3}$ 12)	12+11(14+$\frac{2}{3}$ 12)
Carré	25(101+$\frac{4}{3}$)	11(101+$\frac{4}{3}$)
Inversion	3	3
Couplage	4 (moindre des cas) 7 (pire des cas)	2

TABLE 4.3 – Résultats concrets de la comparaison effectuée dans le tableau 4.2

Il est clair que [117] a un surcoût dans les trois niveaux : Multiplication, Carré et Couplage.

• Deuxième utilisation : Cryptosystème sans couplage, sans Key Escrow, résiste à DoD

Projection de notre approche sur la version 3 du chapitre 2

Setup : Publier < G, K_q, H_1, n, t, > comme paramètres publics.

Le n est le module, H_1 fonction à trappe, K_q corps fini vérifiant certain niveau de sécurité, t paramètre fixe par la PKG de longueur : long(t)=$\frac{3}{4}$long(n)-120-6 (vérifie les conditions de la méthode de génération de e_{ID} citée dans le chapitre 2-partie 2.2.4).

Phase Initiale : génération de la clé de Diffie Hellman

Alice enregistre chez l'autorité 1 (première autorité dans la section 4.2) en stockant $r_{ID_{Alice}}$

Alice choisit un a $\in K_q$, elle envoie directement à Bob :

$$< H_1(s_{Bob} + ID_{Bob}), g^{H_1(s_{Bob})}, g^{s_{Alice} r^{-1}_{ID_{Alice}}}, g^{s_{Alice}}, g^{(s_{Alice}+s_{Bob}+a)} >$$

À la réception, Bob envoie à l'autorité 1 :

$$< g^{s_{Alice} r^{-1}_{ID_{Alice}}}, g^{s_{Alice}}, ID_{Alice} >$$

Autorité 1 vérifie :

$$g^{(s_{Alice} r^{-1}_{ID_{Alice}})^{r_{ID_{Alice}}}} = g^{s_{Alice}}$$

Si l'égalité est vérifiée, l'autorité répond par oui (cad c'est Alice), sinon elle répond par $\perp$

Après une réponse par oui, Bob contacte l'autorité 2 (PKG) pour extraire s_{Bob}

Autorité 2 vérifie tout d'abord l'identité de Bob en calculant

$$g^{H_1(H_1^{-1}(H_1(s_{Bob}+ID_{Bob}))-ID_{Bob})} = g^{H_1(s_{Bob})}$$

Après, elle extrait $g^{s_{Bob}}$, le donne à Bob, ce dernier calcule en toute confiance :

$$g^{(s_{Alice}+s_{Bob}+a)}.g^{-s_{Bob}}.g^{-s_{Alice}} = g^a$$

De la même méthode Bob envoie bP.

Seconde Phase : vérification des clés générées

Alice sélectionne un paramètre m $\in K_q$ privé, elle envoie à Bob :

$$< (g^{mba+s_{Bob}})^{r^{-1}_{ID_{Alice}}}, g^{mba+s_{Bob}} >$$

Après avoir vérifié l'émettrice du message

Bob envoie à Alice :

$$< ((g^{mba})^{b^{-1}}.g^{-a}.g^{s_{Alice}})^{r^{-1}_{ID_{Bob}}}, (g^{mba})^{b^{-1}}.g^{-a}.g^{s_{Alice}} >$$

Le $r_{ID_{Bob}}$ est pour authentifier Bob.

Si :

$$(g^{mba})^{b^{-1}}.g^{-a} = (g^{ma})^{bb^{-1}}g^{-a} = g^{(m-1)a}$$

Alice accepte K = g^{ba} comme vraie clé, sinon elle contacte la PKG.

Encrypt : Alice calcule $e_{ID_{Bob}}$ suivant la méthode 2.2.4 du chapitre 2 :

$e_{ID_{Bob}}$ = convertit en nombre premier(projection sur le code ASCII de ID $\times$ t)

Le message chiffré à Bob est : C=$(M \oplus K)^{e_{ID_{Bob}}}$ mod n.

Extract : Calculé d_{ID} en utilisant la méthode :

$$e_{ID} \times d_{ID} = p_{i_{ID}} q_{j_{ID}} \; mod \; (\varphi(n))$$

Le j_{ID} varie pour chaque identité.

Pour une identité ID_{Bob} extraire $e_{ID_{Bob}} \times d_{ID_{Bob}}$ = $p_i q_j$ mod ($\varphi(n)$).

Decrypt : Calculer $C^{d_{ID_{Bob}}}$ mod n = $M \oplus K$

Après calculer $M \oplus K \oplus K$ =M

Deuxième comparaison

Dans le tableau 4.4, nous comparons le schéma de Yinxia Sun et al [166] et le nôtre (celui du deuxième utilisation).

Ce qui est permanent figure entre setup, KeyDer(s,id), SetUK' pour le schéma de Yinxia Sun et al [166]. Alors que pour le nôtre, le permanent est dans toutes les étapes qui permettent de générer la clé de Diffie Hellman. Le reste pour les deux schémas représente le transitoire.

	Premier tour (r=1)				r tours	
	schéma [166]		le nôtre		schéma [166]	le nôtre
	Perma	Trans	Perma	Trans	-	
Mul	2	3	12	2	-	
Exp	3	9	9	2	-	
Inv	-	-	7	3	-	
Som -me	5Mul+12Exp		11Mul+14Exp +10Inv		(2+3r)Mul+ (3+9r)Exp	(12+2r)Mul+ (9+2r)Exp+ (7+3r)Inv
	-		-		(r=2) 8Mul+ 21Exp	(r=2) 14Mul+ 13Exp+13Inv
	-		-		(r=4) 14Mul+ 39Exp	(r=4) 20Mul+ 17Exp+19Inv
	-		-		(r=6) 20Mul+ 57Exp	(r=6) 24Mul+ 21Exp+25Inv
	-		-		(r=8) 26Mul+ 75Exp	(r=8) 28Mul+ (r=8) 25Exp+31Inv
	-		-		(r=10) 32Mul+ 93Exp	(r=10) 32Mul +29Exp+37Inv
	-		-		(r=12) 38Mul+ 111Exp	(r=12) 36Mul+ (r=12) 33Exp+43Inv

TABLE 4.4 – Comparaison entre notre approche et le schéma de Yinxia Sun et al [166]

Les notations suivantes signifient :

Perma : Permanent

Trans : Transitoire

Analyse des résultas

Tout d'abord, nous notons que le nombre des opérations permanent est constant lorsqu'on utilise notre approche qui est basé sur la génération de la clé de Diffie-Hellman, puisque nous pouvons utiliser ce dernier dès que la clé résultante est non compromise.

Si nous considérons que l'Inverse est moins coûteux que l'Exponentiation, le nombre des

itérations dessinées par le schéma de Yinxia Sun et al [166], s'accroît de plus en plus par rapport à notre. Par exemple, pour r=4, on a :

$$39Exp >> 17Exp + 19Inv \;\; puisque \;\; 17Exp + 19Inv < 36Exp \tag{4.7}$$

On a alors une différence de 3 entre ces deux opérations.
Tandis que pour r=10 on a une différence de 27 et pour r=12 elle est de 36. Cela se voit aussi pour la multiplication où notre schéma ne sera efficace que d'après r=12.

- **Troisième utilisation : Schémas sans révocation, résistant à DoD et sans Key Escrow**

Comme nous l'avons signalé dans le chapitre 1, la révocation des clés présente un problème majeur pour la technologie IBE, cela ne se pose pas pour la PKI puisque la révocation se fait en utilisant un service qui publie l'horodatage des dates et leurs durée de validité, suivant un annuaire connu par CRL (Certificate Revocation List). Afin d'éviter ce problème pour l'IBE, Boneh et Franklin [22] ont proposé de concaténer l'identité à une chaîne de caractères d'horodatage, par exemple, bob@hotmail.fr|15|05|2014. Cette solution parait simple ; mais elle nécessite que la PKG soit online, ainsi, avec le besoin d'une clé sécurisée où il faut la renouveler chaque fois, demande un canal sécurisé entre la PKG et les utilisateurs. Incontestablement, cela fait perdre l'IBE son élasticité. Sonia Jahid et al [110] ont proposé de suivre la méthode proxy proposée par M. Naor et al [141] afin de contourner le problème de révocation des clés dans les Attributs. Dans une procédure de ACM, Boldyreva et al [18] ont proposé une solution considérée comme convenable et où ils adressent directement l'IBE. Cette proposition consiste de générer le système `RIBE`[G] = (`S`, `SK`, `KU`, `DK`, `E`, `D`, `R`) qui est constitué de sept services au lieu de quatre dans le cas de l'IBE.
Le but devant la révocation des clés est de défendre les non autorisés d'accéder aux messages d'un utilisateur dans le cas où leurs clés sont compromises. Dans [18], ainsi que d'autres méthodes, c'est l'autorité qui s'occupe du renouvellement des clés, ce qui la force d'être online et de répondre ainsi à chaque fois aux questions des utilisateurs ou bien les suivre quotidiennement. Par contre, si on ajoute la clé générée par le protocole de Diffie-Hellman aux messages chiffrés, même si la clé générée par la PKG (clé maîtresse dans l'IBE normal) est révoquée, les messages chiffrés seront maintenus puisqu'il y'aura une autre clé fortuite et non compromise.
Comme avec Boneh et Franklin qui proposent une délégation de la part des utilisateurs ce qui leurs donne l'opportunité de les encadrer par eux mêmes, de notre côté nous ferons la

même chose avec la clé résultante mentionnée dans la section 4.2, nous donnerons alors l'occasion aux utilisateurs de renouveler leurs clés par eux-mêmes. En utilisant les cinq algorithmes `S`, `SK`, `KU`, `DK`, , `R` (les `E`, `D` sont souvent utilisés dans le cas normal càd sans intégrer la révocation) l'utilisateur peut renouveler sa clé par lui même sans être lié au PKG. Ceci est avantageux puisqu'il est le seul qui sait quand sa clé est révoquée. Donc, travailler avec la méthode d'IBE normal en renouvelant les clés de Diffie-Hellman par les utilisateurs sera mieux que de charger l'autorité de cette révocation (cas de la méthode [18]).

4.3 Quelques mots sur les attaques physiques

L'essor des attaques physiques dans les technologies fines, donne plusieurs genres sous la position des attaqueurs. Les attaques physiques sont classées en trois catégories : les attaques invasive, les attaques semi-invasive et les attaques non-invasive.
Une attaque invasive est une attaque appliquée physiquement sur le circuit cible et peut conduire à sa destruction. Un peu moins, une attaque semi-invasive agit aussi physiquement sur le circuit mais sans le détraquer, elle exploite les corrélations entre les données exécutées et les signaux. Tandis qu'une attaque non-invasive englobe l'ensemble des techniques qui adressent uniquement les signaux, elle est la plus dangereuse parmi les trois, connue en générale sous le nom d'attaque auxiliaire ou encore à canaux cachés.

4.3.1 Attaques auxiliaires en général

Elles se concentrent sur l'extraction à partir des modules portatifs tels que les cartes à puce, des fuites d'information lors de la mise en oeuvre d'un algorithme contenant des secrets. En utilisant une corrélation qui peut exister entre l'exécution du temps, la puissance de consommation des donnés de radiation (électromagnétique, température, son), ou encore résultats de calcul avec erreurs obtenues par injection de faute lors des implémentations des données secrètes manipulées à l'intérieur d'une carte à puce ou d'un autre circuit (disque dur d'un PC). Les attaques auxiliaires sont composées de six catégories fondamentales. Nous les schématiserons dans la figure suivante :

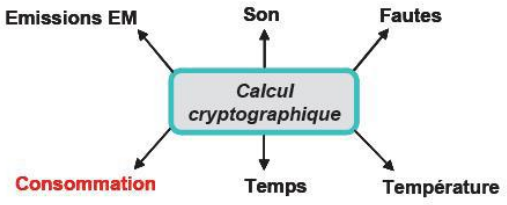


FIGURE 4.2 – Quelques types de l'attaque auxiliaire

Comme figuré en rouge, dans ce chapitre nous scrutons l'attaque par analyse de consommation plus particulièrement l'attaque DPA (acronyme anglais qui signifie Differential Power Analysis).

4.3.2 Attaques DPA en particulier

Les attaques par analyse de consommation sont similaires à peu près au fait que, si nous voulons connaître le succès d'une émission de télévision, nous mesurons seulement la consommation de courant et d'eau lors de sa transmission. Elles consistent en général à étudier les courants et les données courues dans une carte à puce (circuit électronique en général) afin d'extraire des informations secrètes stockées dedans. Les attaques par analyse de consommation se subdivisent en deux : l'attaque par analyse simple de la consommation (SPA) et l'attaque par analyse différentielle de la consommation (DPA).
Le SPA consiste à effectuer une analyse directe des bordures en courant durant les opérations cryptographiques effectuées. Néanmoins, pourqu'elle soit réussie, elle exige une connaissance de l'algorithme cryptographique considéré, surtout de la manière dont il est implémenté.
Une version plus puissante de SPA est l'attaque par analyse différentielle de la consommation (DPA), elle est utilisée lorsque le SPA a echoué. Sa puissance se résume au fait qu'elle ne nécessite aucun détail sur l'implémentation de l'algorithme en cours, elle utilise des analyses statistiques intelligentes (moyen de distance, arrange de coefficients, max de probabilité, etc. ...) qui sont capables d'identifier les plus petites variations de la consommation. C'est pourquoi elle est prise très au sérieux par les promoteurs de circuits sécurisés.
Plusieurs types de l'attaque DPA existent, la plus puissante est HODPA (High Order DPA), puisqu'elle utilise une méthode statique et de corrélation entre différentes entrées. L'attaque DPA est introduite en 1975 par Roland Moreno. Une version plus théorique a été donné par P. C. Kocher, J. Jaffe et B. Jun en 1999 [123]. L'attaque DPA est basée sur une mesure statique de différentes courbes, la mesure de ces courbes se fait comme schématisé ci-dessus (figure 4.3) :

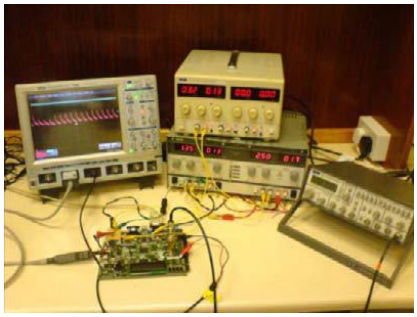


FIGURE 4.3 – Platform utilisée pour mesurer une attaque à consommation

La plupart des circuits électroniques (spécialement ceux de fragment) sont basés sur la technologie CMOS. Dans cette dernière, un changement d'état résulte d'une charge ou d'une décharge électrique des transistors qui sont considérés comme des capacités C. Alors pour une charge d'état d'un bit de 0 à 1, une charge est stockée dans la capacité ce qui signifie que la capacité est connectée à VDD. Pour l'inverse (1-0) la capacité décharge, les états de transitions de bit 0 à 0 ou de 1 à 1 n'influencent pas sur la variation globale d'un circuit électrique.

Principe de l'attaque : version appliquée au couplage

Puisque nous visons d'attaquer les cryptosystèmes de l'IBE, l'attaque DPA qui nous intéresse est celle appliquée au couplage qu'on note ici par e. Pour que l'attaque ait un sens, nous consignons par P le paramètre publique caractérisant un couplage et par R son paramètre secret qu'il faut chercher. Pour exécuter l'attaque, nous faisons une hypothèse sur le bit R, et nous associons selon notre choix des entrées différentes au point connue P. Relançons plusieurs fois l'algorithme de Miller avec ces entrées en mémorisant les traces de courant T_{11},T_{12},..,T_{1m}, (500 $\leq$ m $\leq$ 1000 comme spécifié dans [70]). DPA fait des suppositions sur le secret (hypothèse bit par bit) afin de déterminer la vraie hypothèse sur le secret. Pour l'effectuer nous suivons l'analyse suivante :
Pour une analyse plus claire des donnés, nous assemblons les traces courantes du courant T_{1j} sur deux ensembles dépendant de la valeur du bit b proposé par Miller, nommé bit cible. Nous formons deux ensembles T_0 et T_1 :

$$T_{0j} \in T_0 | Miller(P_j, R)[b] = 0,\ nous\ \ changons\, P_j\ \ selon\, notre\, choix \tag{4.8}$$

$$T_{1j} \in T_1 | Miller(P_j, R)[b] = 1 \tag{4.9}$$

Pour chaque hypothèse, M_0 et M_1, est le moyen respectif des ensembles T_0 et T_1.

$$M_{0i} = \frac{\sum_{j=1}^{m}(1 - Miller(P_j, R)[b]).T_{0i}}{n - \sum_{j=1}^{m}(1 - Miller(P_j, R)[b])} \tag{4.10}$$

$$M_{1i} = \frac{\sum_{j=1}^{m}(Miller(P_j, R)[b]).T_{1i}}{\sum_{j=1}^{m}(1 - Miller(P_j, R)[b])} \tag{4.11}$$

Nous calculons la différence entre les deux moyens : $M^{DPA} = M_1 - M_0$. Si l'hypothèse n'est pas correcte, le bit b est égal à l'actuel bit avec une probabilité $\frac{1}{2}$ pour chaque P_j. Alors la trace résulte du courant par la transition des différents types ([0-1] et [1-0]) peuvent être trouvés

dans les ensembles (T_0 ou T_1). Ainsi, il est naturel que les deux moyens M_0 et M_1 soient égaux, par conséquent, la courbe DPA soit plate et elle tend vers zéro. D'autre part, si l'hypothèse est correcte, le bit b est égal à l'actuel bit avec une probabilité 1. Donc les traces du courant résultent par la transition de même type ([0-1] ou [1-0]) doivent apparaître dans le même type (T_0 ou T_1) et alors une forte amplitude sera affichée.

4.4 IBE et l'attaque DPA, existence et réalité

Récemment, l'attaque DPA appliquée au couplage attire l'attention des chercheurs dans la cryptographie IBE (IBC en général). Puisque 80% de ces crypto-systèmes sont basés sur la notion couplage. Le succès vérifié par l'IBE a donné naissance le 2 Novembre 2004 et pour la première fois dans le monde à l'intégration des crypto systèmes de l'IBE dans les cartes à puce par Gemplus International. Ce qui le laisse susceptible à tous les types d'attaques auxiliaires spécialement l'attaque DPA. Selon notre connaissance, il n'y a aucune étude directe de l'attaque DPA appliquée au crypto systèmes de l'IBE, les études qui existent se limitent à l'attaque du couplage.
Dans les sections qui suivent, nous allons proposer une étude [8] sous forme d'un concept, elle pourra être considérée comme un précède planning pour réussir ce type d'attaques sur l'IBE (IBC en général), notre contribution en ce sens est triple [8].
Nous commençons par donner une proposition convenable pour réussir l'attaque DPA sur les couplages, ensuite, nous exposons des méthodes pour réussir l'attaque DPA sur la cryptographie IBC en général qui sera la première tracée dans la littérature et enfin nous examinons les anciennes contre-mesures en proposant des alternatives convenables.

Tout d'abord, nous lançons un coup d'oeil sur les études reservées à ce type d'attaques appliquées au couplage en examinant leurs validitées.

4.4.1 Visions générales sur les anciennes méthodes faces à l'attaque DPA appliquée au couplage

Les études frimousses à l'attaque DPA appliquées au couplage sont initiées en 2006. La première étude est celle de Dan Page et Frederik Vercauteren [143] où ils ont exploité une étude appliquée à l'algorithme de Duursma-Lee. Malheureusement, leur proposition est non efficace puisqu'elle se limite à l'algorithme Duursma-Lee et à la courbe supersingulière, elle ne développe pas le DPA en global. Claire Whelan et Mike Scott [174] ont fait une étude plus générale, elle peut être appliquer soit à Tate, Ate, Eta, puisqu'ils ont concentré sur

les opérations arithmétiques développées au sein de l'algorithme de Miller : Multiplication (méthode xor), réduction de la racine carrée. La même idée a été utilisée par Tae Hyun Kim et al [119] qui ont utilisé le couplage Eta.

Toutes ces études sont théoriques, une étude plus pratique a été proposée par Nadia El Mrabet et al en 2009 [71] et nous n'avons rencontré aucune étude pratique similaire appliquée au couplage. Nadia El Mrabet et al ont mis en défaut les deux écrits [143] [174]. Cependant notre étude se basera seulement sur celle-ci.

Selon [71], au lieu d'utiliser Miller pour attaquer le couplage, il suffit d'utiliser l'équation de la ligne l_1 (tangente dans Miller) développée par exemple en coordonnée Jacobian (l'attaque fonctionne aussi avec les coordonnées : Affine, Projective, ou encore d'Edward). Afin de valider l'attaque DPA en pratique contre l'algorithme de Miller, Nadia El Mrabet et al ont exploré un circuit où ils ont évalué l_1 utilisé par Miller. L'équatopn de $l_1(x_P; y_P)$ = $Z_3 Z^2 y_P - 2Y^2 - (3X^2 - aZ^4)(Z^2 x_P - X)$ a été divisée en deux portions. Il faut attaquer Z (troisième coordonée de R) à partir de l'étape 2 (R1=$Z^2 \times x_P$), ensuite déterminer X à partir de l'étape 3 (R_1=$Z^2 \times x_P$ - X). Une fois les deux coordonnées X et Z sont déterminées, il sera facile d'attaquer la coordonnée qui reste Y à partir de l'équation de la courbe elliptique.

Malgré sa réussite pratique, cette étude présente les faiblesses suivantes :

— Premièrement, le circuit est simulé sur une architecture de 8 bits et pour un niveau de 160 bits, il faut uniquement le diviser sur une architecture de 8 bits, ce qui est non valide. En effet, on ne peut pas le mesurer sur un DES, puisque, cette dernière contient des SBOX ($SBOX_1 \bigoplus ... \bigoplus SBOX_8$) connecté par XOR, il faut attaquer chaque SBOX dans la dernière étape (16 ronds) qui est chiffré sur 6 bits et faire une recherche exhaustive pour les 8 bits (56-48 = 8) qui restent. Pour l_1 il n'est pas possible de le diviser sur des pièces de 8 bits, il nous faut développer le calcul total pour 160-bits. Ce problème peut être résolu en simulant un circuit qui tient en compte le niveau demandé.

— Deuxième, on ne peut pas baser le succès de l'attaque DPA sur un circuit qui sert cette attaque, il faut baser sur le Software et ne pas construire un serviable Hardware. Un circuit similaire à [71] peut accélérer les calcules dans les cartes à puces, mais du fait du danger de l'attaque DPA, on ne peut pas intégrer ce genre de circuit dans les cartes qui portent des secrets, donc il faut trouver une méthode pour attaquer des cartes contenant un Hardware standard sans oublier que les ID cartes (spécialisées à la cryptographie) peuvent avoir une construction particulière.

— Troisième, le[71] exige de connaître en avance l'ordre r du secret càd celui du point R, ce qui n'est pas possible.

Dans notre proposition nous tiendrons en compte toutes ces faiblesses.

4.4.2 Proposition d'une attaque DPA convenable pour le couplage

Comme il a déjà été souligné, le papier [71] a mis en défaut les deux travaux [143] et [174]. En effet, selon [174] mettre le secret dans le premier paramètre du couplage est une contre-mesure, ce que nous pouvons de notre côté, contredire ; mais d'une manière différente de celle proposée dans [71].

Dans notre étude, nous nous baserons sur l'algorithme de Miller, qui est une étape fondamentale dans le calcul d'un couplage. Donc, attaquer cet algorithme suffit pour attaquer le couplage.

Pour appliquer l'attaque DPA à l'algorithme de Miller, deux cas se présentent : mettre le secret dans le premier argument (1^{er} cas) et après dans le second argument ($2^{ème}$ cas).

• Méthode pour attaquer le premier argument

Dans le **premier cas (**1^{er} **cas)**, puisque le calcul de l'algorithme de Miller est basé sur l'ordre r du premier argument, nous proposons d'attaquer cet ordre, ensuite, de calculer son inverse r'. Attaquer cet ordre nous permet d'attaquer le point en question dans les crypto-systèmes de l'IBE, ce que nous allons le voir plus tard.

Pour attaquer l'odre r, on peut se servir du SPA, mais puisqu'il est simple de lui trouver une contre mesure (SPA est à peu près non efficace aujourd'hui), nous proposons d'utiliser le DPA. Dans l'algorithme de Miller il y a deux étapes différentes, l'étape qui calcule le doublement et celle qui calcule l'addition selon la représentation binaire de l'ordre de r.

Si r_i = 0 (r_i est la représentation binaire de r dans l'étape i) Miller calcule le doublement et si r_i = 1, l'opération d'addition est entrain d'être calculée dans l'algorithme de Miller.

Si nous pouvons distinguer le type de l'étape (addition ou doublement) qui se déroule, nous pouvons alors déterminer le type du bit r (0 ou 1) dans l'étape en question. Pour le faire, nous pouvons bloquer l'algorithme de Miller, par exemple, dans l'étape doublement. Supposons alors r_i=0 où r_i est la représentation binaire de r.

— si f(P) - g(P) = 0 mettre T_i dans T_0.

  Où f est la fonction qu'on peut calculer dans l'étape doublement en utilisant l'algorithme de Miller (un algorithme général, sans utiliser l'algorithme qu'on cherche attaquer son secret). Tandis que g est la fonction qu'on peut retourner initialement par l'algorithme de Miller (qui contient le secret-étape doublement).

— Si f(P) - g(P) = 1 mettre T_i dans T_1.

Calculer M^{DPA} = $M_0 - M_1$, où M_0 et M_1 sont les moyennes des paquets T_0 et T_1 respectivement.

Si la courbe M représente un ou des piques de consommations alors r_i = 0, sinon r_i = 1.

Pour bloquer l'algorithme de Miller dans l'étape doublement ou addition il suffit de connaître

ou au moins simuler le nombre de cycles d'horloges du doublement et celui de l'addition. D'une autre manière, on peut simuler deux points dans une autre carte (expérimentale) : 1^{er} argument et $2^{ème}$ argument, ensuite, appliquer (programmer) l'algorithme de Miller de telle manière qu'il nous affiche par exemple DOU et ADD lorsque nous terminons du doublement et de l'addition respectivement. Bien entendu, le point du premier argument est peut être différent du point R en question (qu'on cherche) ; mais on peut au moins simuler le temps de l'addition et du doublement.
Après cela, il sera simple de préciser le type de l'étape, étape par étape, commençons par 1, 2, 3 ainsi de suite.

• Méthode pour attaquer le second argument

Deuxième cas ($2^{ème}$ cas) : Nous décrivons l'attaque dans ce cas, en admettant qu'il y a plusieurs méthodes pour l'exploiter. Par exemple, pour attaquer le second paramètre, nous avons besoin du calcul de l_1 et v_1 qui sont respectivement la tangente et la verticale dans l'algorithme de Miller. Ces deux lignes ont dans leurs expressions le second paramètre secret corrélé avec celui du premier argument sous une forme simple, alors, tant que ces dernières sont simples (surtout la verticale), mieux on peut parvenir l'attaque DPA.
Le premier paramètre est peut être sous la forme Jacobienne, ceci a un rôle pour accélérer les calculs selon [*15*], le second paramètre peut prendre une forme affine. En utilisant ceci, l'équation de la ligne s'écrit sous la forme :

$$l_1(x_R, y_R) = Z_3 Z^2 y_R - 2Y^2 - (3X^2 - aZ^4)(Z^2 x_R - X) \tag{4.12}$$

Les coordonnées (X, Y, Z) sont celles du premier paramètre (point publique P) qui varient selon notre choix, alors, jouons sur ces choix nous permet d'obtenir de bons résultats. En effet, puisque Z_3 = 2YZ, nous exécutons toujours notre algorithme de Miller pour Y_P = 0, ce qui nous permet d'éliminer la partie qui contient y_R dans l_1, l'équation de ce dernier va changer à :

$$l_1(x_R, y_R) = (aZ^4 - 3X^2)(Z^2 x_R - X) \tag{4.13}$$

Nous notons que prendre toujours Y_P = 0, nous permet d'avoir plusieurs points, puisque nous pouvons choisir n'importe quel Z_P et nous chercherons sur un X_P convenable dans l'équation : $Y^2 = X^3 + aXZ^4 + bZ^6$ qui est sous les coordonnées Jacobiennes (elle est calculée à partir des points $(\frac{X_P}{Z_P{}^2}, \frac{Y_P}{Z_P{}^3})$ qui correspondent aux coordonnées affines).
En suivant cela, pour chaque Z_P choisi on peut obtenir au moins trois X_P convenables à l'équation : $X^3 + aXZ^4 + bZ^6$ = 0. Alors, pour chaque Z_P suggéré, on a : $1 \leq$ nombre(X_P) ≤ 3.

En appliquant l'attaque DPA comme précedemment (partie 4.3.2) à l'algorithme de Miller en combinant le point (X, 0, Z) comme point du premier argument qui varie suivant le temps et nous somme intéressé à l_1 qui a la forme 4.13. Ceci nous permet d'extraire bit par bit le x_R et après, il est simple d'extraire y_R à partir de l'équation : $y^2 = x^3 + ax + b$.
Selon les coordonnées utilisées pour le premier argument et celles du deuxième argument (Affine, Projective, Jacobian ou Edward) on peut avoir d'autres méthodes.

4.4.3 Traduction de notre attaque sur la cryptographie IBC

L'étude exposée dans cette partie sera principalement consacré sur le cryptosystème de Boneh et Franklin [22], l'étude sera aussi valide pour les cryptosystèmes [23] [49] [91] et autres, il faut seulement jouer sur leurs syntaxes.
Avant d'exploiter cela, nous notons que pratiquer les cryptosytèmes de l'IBE dans les cartes à puces est plus envié, puisque la clé publique est sous forme d'une identité, par contre, dans la PKI il faut mettre en service une façon pour extraire la clé publique.
Les cartes à puces peuvent être aussi utilisées en IBE pour stocker quelques paramètres afin de les réutiliser. En effet, pour une communication continue entre l'émetteur et le récepteur, le premier peut réutiliser, par exemple, le schéma de Boneh Franklin [22] pour chiffrer les messages au récepteur en utilisant toujours les mêmes paramètres publiques :

$$< q, G_2, G_2, e, n, P, P_{pub} = sP, Q_{ID}, H_2 >$$

La seule chose que l'émetteur change est r qu'il modifie pour chaque message (puisque C = ciphertext = $< rP, M \bigoplus H_2(g^r) >$ = $< U, V >$, où r est un paramètre choisi par l'émetteur). Alors pour ne pas le recalculer à chaque fois, l'utilisateur programme seulement la fonction r $\longrightarrow$ g^r = $e(Q_{ID}, P_{pub})^r$ et le stock quelques part. La bonne place pour le cacher afin de réutiliser les calculs en changeant seulement r, est dans une carte à puce. Or, celle-ci demande impérativement d'y accéder, ce qui n'est pas possible. La seule chose qui reste à la portée de l'ennemi est d'utiliser les canaux cachés et plus particulièrement l'attaque DPA.
Pour programmer g^r trois opportunités sont offertes à l'utilisateur (l'émetteur) :

1. Utiliser r dans le premier argument (g^r = $e(rQ_{ID}, P_{pub})$)
2. Dans le second argument g^r = $e(Q_{ID}, rP_{pub})$
3. Ou encore dans l'exposant g^r = $e(Q_{ID}, P_{pub})^r$

Ceci pour la communication (émetteur-récepteur) càd celle du chiffrement. En ce qui concerne le déchiffrement, du fait de r qu'on change à chaque fois, il est possible que le récepteur ait besoin du calcul : $e(d_{ID}, U)$ = $e(sQ_{ID}, rP)$ (ou de $e(U, d_{ID})$ = $e(rP, sQ_{ID})$) pour le réutiliser, il le stock dans une carte à puce.

Nous allons examiner ces trois cas de chiffrement (le cas du déchiffrement est peut être mesuré sur eux) afin de savoir la plus sensible à l'attaque DPA et donc de l'éviter.

S'agissant du premier cas du chiffrement, le secret est dans le premier argument et alors selon notre méthode citée dans la partie 4.4.2-celle du premier argument, il suffit d'attaquer l'ordre r' de rQ_{ID}. Attaquer r' en connaissant l'ordre $r_{Q_{ID}}$ du point Q_{ID}, nous permet d'attaquer r puisque r=$r_{Q_{ID}}r'^{-1}$. En réalité, $r_{Q_{ID}}$ est non connu ; mais puisque Q_{ID} est publique, on peut le calculer en utilisant l'algorithme qui suit (on peut aussi utiliser l'algorithme NAF ou autres méthodes) :

Algorithme 4.4.1 pour calculer la multiplication scalaire

Entré : a = m, B = O, C = Q_{ID} ;

1. Si a est pair a $\leftarrow \frac{a}{2}$, B=B, C=2C ;
2. Si a est impair, a $\leftarrow$ a-1, B=B+C,C=C ;
3. Si $a \neq 0$, aller a l'étape 2.

Sortir B

Remarques 4.4.1 :

1. *Pour calculer $r_{Q_{ID}}$ on peut commencer par m=q, puisque $Q_{ID} \in G_1$ qui est cyclique d'ordre q. Avec bQ_{ID} (b < q est de notre choix, on peut le choisir plus grand), on peut accélérer l'algorithme 4.4.1, nous demandons à l'algorithme d'afficher t dans la sortie lorsque bQ_{ID}=O et alors $r_{Q_{ID}} = tb^{-1}$.*
2. *On ne peut pas attaquer r à partir de rQ_{ID}, rP ou une autre expression, puisqu'il s'agit du problème de logarithme discret.*
3. *L'autorité peut mettre à la disposition de l'utilisateur $r_{Q_{ID}}$, puisque l'émettrice Alice en aura besoin pour calculer, par exemple, $e(rQ_{ID}, P_{pub})$ (pour calculer l'ordre de rQ_{ID}, Alice besoin de connaître l'ordre de Q_{ID}).*

Une fois r est attaqué, il sera simple de calculer $e(Q_{ID}, P_{pub})^r = g^r$.

Pour ce qui est du second cas du chiffrement càd $e(Q_{ID}, rP_{pub})$, notre méthode du second cas, présentée dans le partie 4.4.2-celle du second argument, nous permet d'attaquer directement rP_{pub} et alors on peut calculer facilement $e(Q_{ID}, rP_{pub}) = g^r$.

Quand au troisième cas du chiffrement càd celui lié à l'exposant (g^r = $e(Q_{ID}, P_{pub})^r$). Pour calculer l'exponentiation, plusieurs méthodes peuvent être utilisées, on cite celle proposée par Chaine (addition) ou de Windows, les deux sont sensibles à l'attaque par consommation. Par exemple, pour la première càd par Chaine, l'attaque DPA est efficace pour trouver le secret (l'étude de Jean-Sebastien Coron [57]) et pour celle de Window, seulement l'attaque SPA est suffisante (Pierre-Alain Fouque et al [77]). Par soucis qu'il y a plusieurs méthodes qui peuvent exploiter l'exponentiation, on ne peut ni expliquer ces méthodes dans ce mémoire ni intégrer aucune autre proposition ; et puis, nous nous limiterons de donner des contres mesures qui lui sont convenables.

Revenons maintenant au déchiffrement, dans $e(sQ_{ID}, rP)$ le secret figure dans le premier argument, il est d_{ID} puisque le second point rP est publique. L'attaque de $e(sQ_{ID}, rP)$ est peut être faite de la même manière que pour le 1^{er} cas du chiffrement càd attaquer l'ordre r' de sQ_{ID} et en utilisant $r_{Q_{ID}}$ on peut déterminer s=$r_{Q_{ID}}$-r', ce qui est présente un danger puisque s est la clé maîtresse.
L'autre cas $e(rP, sQ_{ID})$ se traite en suivant la méthode du second argument.

En fait, attaquer g^r présente une menace à la communication Alice-Bob, en plus, il nous permet de calculer H_2(g^r) puisque H_2 est publique. Ce qui permet à l'adversaire Eve d'attaquer l'un des messages de la communication Alice-Bob, après qu'elle ait calculé :

$$M \oplus H_2(g^r) \oplus H_2(g^r) = M$$

Eve peut alors suivre la communication Alice-Bob, dans ce cas, cette adversaire est proche d'Alice ou de Bob. Elle est leur collègue ou bien leur cliente de travail.

Aujourd'hui les cartes à puces ne peuvent pas opérer sur Internet, et donc à part l'idée de stocker g^r pour le réutiliser, cette étude s'adresse aux cartes à puces bancaires ou encore téléphoniques dont l'IBE est utilisé pour cacher les secrets. On a exploité notre idée sur le schéma basique de Boneh et Franklin même s'il n'est pas CCA2 sécurisé, ceci afin de simplifier la démonstration. La même méthode peut être appliquée sur sa version complète.

En réalité, les cartes à puces sont utilisées pour cacher la signature ou un protocole de ce qu'on appelle Key agreement. En effet, il se peut qu'une autorité signe la clé privée de l'émetteur à un récepteur dans une carte à puce. Nous avons examiné toutes les signatures (de l'IBC) : Sakai-Ohgishi-Kasahara's ID-based signature (IBS) [155], Hess's IBS [100], Cha-

Cheon's IBS [48], Paterson's IBS [144], nous notons qu'elles ne donnent pas de bons résultats face à l'attaque DPA. En effet, le grand but qu'on imagine devant l'attaque d'une signature est d'attaquer la clé secrète S_{ID}, cela est sûrement impossible d'après les formules de [155] [100] [48] [144], en plus, le fait que la signature est utilisée une seule fois permet de défendre ce type d'attaque. Par contre, le DPA peut influencer sur le protocole du Key agreement, puisque par exemple, si nous prenons le schéma de Chen et Kudla (Key Agreement) [54], il est possible qu'une entité A stock une clé K_{AB} = $e(S_A, T_B + aQ_B)$ dans une carte à puce. Mais attaquer en même temps S_A et $T_B + aQ_B$ est difficile. En effet, si nous changeons l'une des entrées pour réaliser une attaque DPA nous perdrons le secret. Alors soit qu'on attaque S_A qui est la clé secrète en utilisant le 1^{er} cas du chiffrement, ou qu'on attaque $T_B + aQ_B$ en utilsant $2^{ème}$ cas du chiffrement et après avoir calculé $T_B + aQ_B - T_B$ (puisque $T_B = bQ_B$ est publique). Mais attaquer K_{BA} = $e(Q_A, Q_B)^{s(a+b)}$ parait difficile, sauf si nous avons l'opportunité d'obtenir deux cartes, une qui contient K_{AB} = $e(S_A, T_B + aQ_B)$ et l'autre contient K_{BA} = $e(T_A + bQ_A, S_B)$. Cela peut se faire par une coopération entre deux adversaires, l'un (Eve) à l'opportunité d'obtenir K_{AB} tandis que l'autre (Cesar) peut obtenir la carte qui contient K_{BA}. Alors Eve peut accèder à s après avoir attaqué S_A = sQ_A en utilisant la méthode du 1^{ere} cas de chiffrement, de la même méthode César peut calculer a + b après avoir accèdé à $T_A + bQ_A$ = (a+b)Q_A. Alors la clé : K = $kdf(K_{AB})$ = $kdf(K_{BA})$ = kdf($e(Q_A, Q_B)^{s(a+b)}$) peut être extraite facilement, kdf est la clé de dérivation de la fonction (kdf est peut être une fonction de hachage H_2 :$G_2 \longrightarrow \{0,1\}^*$).
La même méthode peut être appliquée aux autres protocoles.

Jusqu'ici, nous avons présenté notre vision pour l'attaque DPA appliquée à la cryptographie IBC, nous allons présenter sous quel niveau on peut arriver, en comparant notre méthode avec celle de Nadia El Mrabet et al [71] qui est purement pratique.
Pour des raisons pratiques, nous référençons nos deux méthodes : placer le secret dans le premier argument et dans le second argument avec l'étude de [71] qui adresse uniquement le premier argument. Notre comparaison a pris en considération le nombre de traces de chaque méthode, nous ferons la comparaison en oubliant qu'il faut augmenter le nombre de traces lorsqu'une courbe non désirable est affichée (petite amplitude). Selon [70], l'étude peut être réussie si on la corrèle avec des nombres publiques entre 500 et 1000 pour la multiplication [70] et un nombre de 65 280 = $2^9 5^2$ pour la soustraction [70]. Nous acceptons dans ce qui suit 800 = $2^5 5^2$ choix pour les deux études [71] et la nôtre, aussi, nous acceptons un niveau de 160 bits pour les deux points P (publique) et R (secret).

Étude [71]	nb_{traces} d'attaquer Z_R	nb_{traces} d'attaquer X_R	nb_{traces} d'attaquer l_1=Somme
Un seul bit	$2^5 5^2$ traces	$2^8 \times 5 \times 51$ traces	$2^5(5 + 2^3 \times 51)$ traces
160 bit	$2^{165} 5^2$ traces	$2^{168} \times 5 \times 51$ traces	$2^{165} \times 2065$ traces

Notre [8]	nb_{traces} d'attaquer 1^{er} argument (ordre r)	nb_{traces} d'attaquer $2^{ème}$ argument (x_R)
Un seul bit	$2^5 5^2$ traces	$2^8 \times 5 \times 51$ traces
160 bit	$2^{165} 5^2$ traces	$2^{168} \times 5 \times 51$ traces

TABLE 4.5 – Comparaison entre notre étude [8] et celle de Nadia El Mrabet et al [71] selon le nombre de traces

nb_{traces} signifie nombres de traces.
Il est visible que :

$$2^{165}5^2 = 2^{165} \times 25 << 2^{165} \times 2065 \quad et \quad que \quad 2^{168} \times 5 \times 51 = 2^{165} \times 2040 << 2^{165} \times 2065.$$

4.4.4 Vers des meilleures contre-mesures

Une contre mesure permet de défendre les attaques à canaux cachés plus particulièrement l'attaque DPA. Elle est divisée en deux : Hardware et Software. Dans la Hardware, on fait des changements au sein du circuit (ajouter des OR) pour défendre une attaque quelconque. Alors qu'en Software on s'intéresse à la théorie. Dans le présent chapitre, nous examinons uniquement le Software.

Plusieurs contre-mesures (sous forme Software) sont proposées pour défendre l'attaque DPA face au couplage, selon nos connaissances nous citons :

— La proposition de J.S.Coron [57] :

$$\forall \lambda \in {F_p}^*, (X, Y, Z) = (\lambda^2 X, \lambda^3 Y, \lambda Z) \tag{4.14}$$

— Celle de D.Page et F.Vercauteren [143] :

$$e([s]P, [r]Q) = e(P, Q)^{sr} \; tel \; que \; sr = 1 \; mod(l) \; (l \; est \; l'ordre \; de \; l'algorithme \; de \; Miller) \tag{4.15}$$

— L'autre de C.Whelan et M.Scott [174] :

$$e(P, Q) = e(P, Q + R)e(P, R)^{-1}. \tag{4.16}$$

Dans ce qui suit, nous examinons ces trois contres-mesures, mettant en défaut quelques unes et proposant des alternatives convenables.

• Étude des anciennes contres-mesures

Commençons par la première (celle de Coron [57]), cette contre-mesure est basée sur l'homogénéité, alors (λ^2X,λ^3Y,λZ) peut jouer le même rôle que (X, Y, Z).
Nous confirmons que cette contre-mesure est fragile, en effet, si on attaque (λ^2X,λ^3Y,λZ), on peut attaquer façilement (X, Y, Z), d'après le fait que :

$$\frac{\lambda Z}{\lambda^2 X} \times \lambda Z = cte_0 \Longrightarrow X = cte_1 Z^2 \tag{4.17}$$

Aussi :

$$\frac{\lambda^2 X}{\lambda^3 Y} \times \lambda Z = cte_2 \Longrightarrow Y = cte_3 XZ \Longrightarrow Y = cte_1 \ cte_3 \ Z^3 \tag{4.18}$$

Nous remplaçons dans l'équation $Y^2 = X^3 + aXZ^4 + bZ^6$ (équation de la courbe elliptique mise en jeu), X et Y par les expressions 4.17 et 4.18, cela nous permet d'extraire Z et alors X et Y en utilisant toujours 4.17 et 4.18. Par conséquent, on peut mettre en défaut les précautions qui disent que cette contre mesure est plus rigide.
Par contre, la contre mesure e([s]P,[r]Q) = $e(P,Q)^{sr}$ = e(P,Q), peut rendre le service puisque nous ajoutons r,s tel que sr = 1 mod l pour masquer P et Q. Mettre le secret dans P ou Q et le masquer par r ou s paralyse l'attaque. En effet, même si nous attaquons le secret où paraît dans son expression s ou r, il nous reste le second paramètre qui est aussi masqué. Il n'est pas possible d'attaquer le secret masqué qui est stocké dans un paramètre, en même temps, extraire le masque du second paramètre. Or, pour attaquer le secret dans un paramètre, il faut changer l'autre argument, et une fois ce changement est fait, nous perdons l'autre argument. En outre, l'idée [174] selon laquelle, prendre n'importe r et s tel que rs=1 mod l peut cheminer la contre-mesure n'est pas juste, puisque, on ne peut pas obtenir le même r et s stockés dans la carte à puce.
En guise de conclusion, cette contre-mesure paraît convenable pour défendre l'attaque DPA.
En ce qui concerne la troisième contre-mesure, nous doutons de son éfficacité. En effet, on a : e(P,Q) = e(P,Q + R)$e(P,R)^{-1}$ donc le secret est dans le second paramètre masqué par R (le secret est Q, si c'est P la contre mesure n'a pas de sens) et nous retranchons ce masque par $e(P,R)^{-1}$. Cette expression a un inverse, et puisque ce dernier consomme plus électriquement on peut différencier e(P,Q+R) de e(P,R). Alors, on peut attaquer premièrement Q+R et après on attaque R, on calcule ensuite Q+R-R = Q.
Maintenant qu'on a traité l'efficacité de chaque contre-mesure, nous traitons dans la suite leur influence sur nos méthodes qui sont citées dans la partie 4.4.2, en proposant des alternatives convenables.

• Discussion et proposition des alternatives convenables

Aucune des trois contres mesures n'adressent l'exponentiation qui figure dans l'expression des couplages. Les deux couplages Tate et Weil, se réduisent au module l, puisque, la sortie de Weil est μ_l (l'ensemble de tout les $l^{ième}$ racines de l'unité), celle de Tate est $\frac{k^*}{(k^*)^l}$. Alors pour les deux couplages on a : $e(P,Q)^l$ = e(P,Q), tilisant ceci, on peut construire une contre-mesure. Choisissons λ un nombre arbitraire, alors :

$$e(P,Q)^r = e(P,Q)^{r+\lambda l}$$

On ne peut pas extraire r puisqu'on ne sait pas λ et de préférable l, nous conseillons d'avoir un l inconnu. Une autre contre-mesure qu'on peut construire contre l'attaque de l'exponentiation est de choisir λ et λ' tel que : $\lambda\lambda'$ = 1 et calculer après $e(P,\lambda Q)^{\lambda' r}$ = e(P,Q). Alors, on ne peut pas extraire r puisqu'on ne connait pas λ', en plus, λQ est placé dans le second argument. Extraire λ demande de casser le logarithme discret.

Cela d'une part, d'autre part, pour défendre nos méthodes (mettre le secret dans le premier argument ou dans le second argument), nous avons vu que la contre mesure :
e([s]P,[r]Q) = $e(P,Q)^{sr}$ tel que : rs=1modl, peut rendre le service.
En ce qui concerne la mise du secret dans le premier argument, nous proposons en plus, la cache e(σP,Q) où σ est un paramètre qu'on peut ajouter dans les paramètres publics des schémas d'IBE, par exemple, dans celui de Boneh et Franklin :

$$< q, G_2, G_2, e, n, P, P_{pub} = sP, Q_{ID}, \sigma Q_{ID}, H_2 >$$

Le σ construit est inversible d'inverse σ', il suffit donc pour déchiffrer de calculer $e(\sigma P,Q)^{\sigma'}$. Cela est évidement efficace pour défendre notre méthode qui consiste à trouver l'ordre du paramètre secret. En effet, le secret dans Boneh et Franklin est r qui figure dans l'expression du chiffrement : $e(rQ_{ID}, P_{pub})$, par notre méthode proposée dans la partie 4.4.3, on peut attaquer l'ordre de rQ_{ID} et on peut ainsi déchiffrer les messages ; mais si on change cette expression par $e(r\sigma Q_{ID}, P_{pub})$, la seule chose qu'on puisse attaquer est l'ordre de $r\sigma Q_{ID}$ et on ne peut nul part attaquer celui de rQ_{ID} comme on ne connait pas σ.

Dans le cas où le secret est dans le second argument, la contre mesure : e(P,Q + R)$e(P,R)^{-1}$ coûte chère pour une carte à puce, en plus, nous doutons de son efficacité. Pour cette raison, nous proposons d'utiliser la cache e(P,Q+aP), puisque e(P,Q+aP)=e(P,Q). On a : e(P,aP)=1, avec P publique, le a est un paramètre secret choisi par celui qui désire apporter la contre mesure.

4.5 Attaque par faute face au couplage, vision générale et clairvoyance sur sa sécurité

4.5.1 Vision générale

• Définition et principe

L'attaque par injection de fautes est comptée parmi la catégorie semi-invasive, son principe consiste à générer d'une manière exprès des fautes dans un algorithme (schéma en général) au cours de son exécution afin d'obtenir des comportements anormaux. Les injections de fautes peuvent être excitées en utilisant un laser, des radiations électromagnétiques ou simplement un changement de température. Deux types de fautes peuvent être générées, permanentes ou transitoires.
En ce qui concerne les fautes permanentes, la valeur d'une cellule mémoire a définitivement changé, elle induit une erreur d'une manière permanente. Par contre, lors des fautes transitoires, seulement, l'exécution d'un code ou d'un calcul intégré qui est perturbée, d'une manière temporaire.

• Evocation rapide des anciennes propositions

Dans [143], Page et Vercauteren proposent une attaque par fautes contre l'algorithme de Duursma et Lee. Leur attaque consiste à perturber le nombre d'itération de cet algorithme sous les coordonnées Affines. Dans [69], Nadia El Mrabet a développé leur idée pour qu'elle satisfasse l'algorithme de Miller, sa proposition nécessite d'avoir deux résultats consécutifs dans l'étape du doublement ou d'addition, qui soient $f_{\tau,P}(Q)$ et $f_{\tau+1,P}(Q)$, et calculer ensuite le rapport R=$\frac{f_{\tau+1,P}(Q)}{f_{\tau,P}(Q)^2}$. L'attaque repose sur la résolution d'un système obtenu par identification des éléments dans la base de F_{p^k}. En utilisant les cordonnées jacobiennes et k=4, l'auteur a trouvé un système simple si $r_{\tau+1} = 0$ et un peu difficile si $r_{\tau+1} = 1$.
Dans [175] les auteurs ont généralisé l'idée mentionnée dans [69] pour qu'elle satisfasse tout les degrés de plongement k de parité paire. En utilisant ce genre de k, les auteurs ont montré que quelque soit la position de l'argument, on peut attaquer le secret, ainsi, ils prouvent que le cas où $r_{\tau+1} = 1$ ne présente aucune difficulté.
Mal-à-propos, le [143] est seulement valable pour l'algorithme Duursma et Lee où figure le produit

$$\prod_{i=1}^{m}[(-{y_P}^{3^i}.{y_Q}^{1/3^{i-1}}\sigma.(x_P^{3^i}+x_Q^{1/3^{i-1}}+b)^2).(x_P^{3^i}+x_Q^{1/3^{i-1}}+b)^2)\rho-\rho^2] \tag{4.19}$$

Donc lorsque nous obtenons deux résultats après une injection par faute $R_{m'+r}$ et $R_{m'+r+1}$, leur rapport se simplifie à :

$$(-{y_P}^{3^i}.{y_Q}^{1/3^{i-1}}\sigma.(x_P^{3^i}+x_Q^{1/3^{i-1}}+b)^2).(x_P^{3^i}+x_Q^{1/3^{i-1}}+b)^2)\rho-\rho^2. \quad (4.20)$$

Il est alors usuel d'après la forme de 4.20 que ce n'est pas simple d'attaquer ni x_P ni x_Q. Plus que cela, pour les couplages où subsistent une exponentiation (Tate, Ate, Eta, Twisted ate..), réussir l'attaque nécessite d'inverser q^k-1 et connaître la racine en question, ou k est un degré de plongement choisi. Le problème est traité pour k=3, k=6 dans [112] et [174] respectivement ; mais pour des degrés généraux, cela pose un problème, c'est pourquoi l'exponentiation se compte comme une contre mesure. Ce problème ne se pose pas pour Nadia el Mrabet [69], puisqu'il y a de nombreuses méthodes dans la littérature microélectronique qui permettent d'arrêter les calculs avant l'exponentiation, lire le résultat intermédiaire entre l'exécution de l'algorithme de Miller et l'exponentiation, ou encore annuler l'étape d'exponentiation. Malheureusement, les deux études [69] et [175] requièrent impérativement d'avoir deux résultats consécutifs. Ce qui n'est pas toujours possible.
Nous admettons que la probabilité P(n,N) = 1 - $\frac{B(n,N)}{C_{n+N}^n}$ posée pour obtenir ces résultats consécutifs conduit a bien. Dans la partie ci-dessous (càd 4.5.2), nous étudions la validité des traditionnelles contres-mesures face aux propositions [143], [69] et [175].

4.5.2 Clairvoyance sur la sécurité des attaques par fautes reservées au couplage

Santosh Ghosh et al, ont prouvé dans [94] que les deux contre-mesures créées dans [143][174] ne suffisent pas pour défendre l'attaque par faute face à l'algorithme de Duursma-Lee. La raison réfère au fait qu'elles ne touchent pas les itérations internes et la finale m. Les contre-mesures proposées changent seulement l'entrée sans influencer sur la sortie. Donc même si nous utilisons une des contre-mesures proposées dans [143][174], nous obtenons les mêmes sorties $R_{m'+r}$ et $R_{m'+r+1}$ que le couplage original, après avoir appliqué l'injection par fautes. Par exemple pour la contre-mesures e(xP,yQ) (xy=1 mod l), en utilisant

$$\frac{R_{m'+r+1}}{R_{m'+r}}=(-{y_{xP}}^{3^i}.{y_{yQ}}^{1/3^{i-1}}\sigma.(x_{xP}^{3^i}+x_{yQ}^{1/3^{i-1}}+b)^2).(x_{xP}^{3^i}+x_{yQ}^{1/3^{i-1}}+b)^2)\rho-\rho^2 \quad (4.21)$$

et l'équation de la courbe elliptique prise en considération, nous pouvons extraire facilement le paramètre secret P. Pour défendre cela, les auteurs ont ajouté des itérations au sein de l'algorithme et ils ont prouvé l'efficacité de leur apport. Santosh Ghosh et al ont ensuite exa-

miné l'effet de l'attaque par fautes sur l'algorithme de Miller et ils se sont restreint à le juger sous les coordonnées Edwards. Ils sont arrivés à conclure que le couplage de Tate $e_r(P,Q)$ est sensible pour l'itération r=2 et que les contre-mesures proposées dans la littérature (qui sont aussi citées dans la partie 4.4.4) ne peuvent pas bloquer l'attaque, ils ont ensuite proposé une contre-mesure convenable. De notre côté, nous avons examiné l'attaque proposée dans [69] et [175], pour les paralyser nous proposons [8] d'amener à l'algorithme de Miller le changement suivant :

Petit changement dans Miller pour bloquer l'attaque DFA quel que soit le degré de plongement pair

Pour bloquer l'attaque [175] (valable aussi pour l'étude [69] qui est un cas particulier de [175] puisqu'elle présente k=4), nous proposons [8] d'ajouter un entier aléatoire r_2 (choisi dans le corps F_p) dans l'algorithme de Miller (voir chapitre 1) comme élaboré dans l'algorithme ci-dessous.

Algorithme 4.5.1 : Modifié de Miller(P, Q, r) *($f_{r,P}(Q) \in G_3$ ($\subset {F_{p^k}}^*$))*

Entrée : r = (r_n**...**r_0**)**(représentation binaire),

P $\in G_1$ ($\subset E(F_p)$) et **Q** $\in G_2$ ($\subset E'(F_{p^k})$).

T $\leftarrow$ P, $f_1 \leftarrow 1$, $f_2 \leftarrow 1$

Choisir $r_2 \in F_p$

Pour i = n - 1 à 0 **faire**

T $\leftarrow$ [2]T

$f_1 \leftarrow {f_1}^2 \times l_1(Q)\times r_2$

l_1 est la tangente à la courbe en T.

$f_1 \leftarrow f_1 \times v_1$(Q) $\times r_2$

v_1 est la verticale à la courbe en [2]T.

si r_i=1 **alors**

$f_2 \leftarrow {f_2}^2 \times l_2(Q)\times r_2$

l_2 est la droite qui passe par (PT).

$f_2 \leftarrow f_2 \times v_2(Q)\times r_2$

v_2 est la verticale au point P + T.

Fin si.

Fin pour.

Retourner $\frac{f_1}{f_2} \times r_2^{(numi=0)-(numi=1)-1}$

Avertissement 4.5.1 : L'algorithme modifié de Miller est exact et il emmène au même résultat que le traditionnel algorithme peut achever, après l'exponentiation finale (couplage de Tate et ses variantes).

Preuve :

Puisque $r_2 \in F_p$, après avoir élevé à la puissance $\frac{p^k-1}{r}$, le calcul se simplifie au calcul original or $r_2^{\frac{p^k-1}{r}} = 1$. En effet :

$$\frac{(p^k-1)}{r} = \frac{(p^k-1)}{\phi_k(p)} \times \frac{\phi_k(p)}{r} \tag{4.22}$$

(résultat de Koblitz et Menezes [122]) et puisque :

$$p^k - 1 = \prod_{k'/k} \phi_{k'}(p) \tag{4.23}$$

Donc :

$$\phi_1(p) = p - 1/\frac{(p^k-1)}{\phi_k(p)} \tag{4.24}$$

□

Ce résultat parait valable au couplage où figure l'exponentiation (Tate et ses variantes) et dont celui de Weil ne peut pas bénéficier. Mais il est possible d'élever ce dernier à une puissance et il est prouvé dans [176] que cette opération est aussi utile pour réduire la complexité du couplage de Weil.

Avertissement 4.5.2 : L'algorithme modifié de Miller résiste à l'attaque par faute présentée dans [175] (aussi à [69]).

Preuve :

Nous limitons à faire le jugement lorsque le secret est placé dans le premier argument
(et dans le cas des coordonnées Jacobiennes), c'est un cas plus difficile à résoudre, bien entendu, lorsque le secret est placé dans le second argument l'attaque ne demande que quelques simples déductions.

Commençons par le cas où $r_{\tau+1} = 0$, après l'identification et utilisant les deux équations :

$$R = \frac{f_{\tau+1(Q)}}{{f_{\tau(Q)}}^2} \tag{4.25}$$

et

$$f_{\tau+1,P}(Q) = (f_{\tau,P}(Q))^2(2{Z_j}^3 Y_j y\sigma - 2{Y_j}^2 - 3({X_j}^2 - {Z_j}^4)(x{Z_j}^2 - X_j) \tag{4.26}$$

Avec le fait que :

$$R = R_{2n-1}\xi^{n-1}\sigma + R_{2n-2}\xi^{n-2}\sigma + ... + R_n\sigma + R_{n-1}\xi^{n-1} + R_1\xi + R_0) \tag{4.27}$$

Les deux équations 4.25 et 4.26 conduisent au système suivant :

$$\begin{cases} 2Z_j^3 Y_j y_{n-1} = R_{2n-1} \\ 2Z_j^3 Y_j y_{n-2} = R_{2n-2} \\ . \\ . \\ 2Z_j^3 Y_j y_0 = R_n \\ (-3Z_j^2(X_j^2 - Z_j^4))x_{n-1} = R_{n-1} \\ . \\ . \\ -3Z_j^2(X_j^2 - Z_j^4)x_1 = R_1 \\ -3Z_j^2(X_j^2 - Z_j^4)x_0 + 3(X_j^2 - Z_j^4)X_j - 2Y_j^2 = R_0 \end{cases} \tag{4.28}$$

Après avoir remplacé les expressions de x et y respectivement par : $x_0 + x_1\xi + ... + x_{n-1}\xi^{n-1}$ et $y_0 + y_1\xi + ... + y_{n-1}\xi^{n-1}$ et en effectuant la mutation proposée dans l'algorithme modifié de Miller au système 4.28, ce dernier va changer à :

$$\begin{cases} 2r_2 Z_j^3 Y_j y_{n-1} = R_{2n-1} \\ 2r_2 Z_j^3 Y_j y_{n-2} = R_{2n-2} \\ . \\ . \\ 2r_2 Z_j^3 Y_j y_0 = R_n \\ (-3r_2 Z_j^2(X_j^2 - Z_j^4))x_{n-1} = R_{n-1} \\ . \\ . \\ (-3r_2 Z_j^2(X_j^2 - Z_j^4))x_1 = R_1 \\ r_2(-3Z_j^2(X_j^2 - Z_j^4)x_0 + 3(X_j^2 - Z_j^4)X_j - 2Y_j^2) = R_0 \end{cases} \tag{4.29}$$

$$\Downarrow$$

$$\begin{cases} 2r_2 Z_j^3 Y_j y_{n-1} = R_{2n-1} \\ 2r_2 Z_j^3 Y_j y_{n-2} = R_{2n-2} \\ . \\ . \\ Z_j^3 Y_j = \lambda_0 r_2' \ [4.a] \\ (-3Z_j^2(X_j^2 - Z_j^4))x_{n-1} = R_{n-1} \\ . \\ . \\ Z_j^2(X_j^2 - Z_j^4) = \lambda_1 r_2' \ [4.b] \\ -3Z_j^2(X_j^2 - Z_j^4)x_0 + 3(X_j^2 - Z_j^4)X_j - 2Y_j^2 = \lambda_2 r_2' \ [4.c] \end{cases} \tag{4.30}$$

Où r_2' représente l'inverse de r_2 dans F_p (les deux sont inconnus). De [4.a] on tire le fait que $Y_j = \frac{\lambda_0 r_2'}{Z_j^3}$ (eq 1), le [4.b] permet d'avoir $X_j^2 - Z_j^4 = \frac{\lambda_1 r_2'}{Z_j^2}$. Ensuite, en utilisant ces deux équations ainsi que [4.c] conduisent à l'équation : $X_j = \frac{(\lambda_2 + 3\lambda_1 x_0){Z_j}^6 + 2\lambda_0 r_2'}{3\lambda_1 {Z_j}^4}$ (eq 2). En substituant cette équation dans [4.b], nous permet d'obtenir :

$$({\lambda_2}^2 + 9{\lambda_1}^2{x_0}^2 - 9{\lambda_1}^2){Z_j}^{12} + (4{\lambda_0}^2\lambda_2 r_2' + 12{\lambda_0}^2\lambda_1 x_0 r_2' - 9{\lambda_1}^3 r_2'){Z_j}^6 + 4{\lambda_0}^4 {r_2'}^2 = 0 \ (eq\ 3). \tag{4.31}$$

La dernière équation (eq 3) ne peut pas être résolue puisqu'elle contient deux inconnues Z_j et r_2'. La même chose sera dite sur les deux autres ((eq 1) et (eq 2)) où figure r_2'. Comme conclusion, on ne peut nul part extraire le secret.

Concernant le cas où $r_{\tau+1}$ = 1, l'attaque repose sur les deux équations :

$$R = \frac{f_{\tau+1(Q)}}{{f_{\tau(Q)}}^2} \tag{4.32}$$

et

$$\begin{aligned} f_{\tau+1,P}(Q) = (f_{\tau,P}(Q))^2(2{Z_j}^3 Y_j y\sigma - 2{Y_j}^2 - 3({X_j}^2 - {Z_j}^4)(x{Z_j}^2 - X_j)) \\ (Z_{2j}(X_p{Z_{2j}}^2 - X_{2j})y\sigma) - (Y_p{Z_{2j}}^3 - Y_{2j})x - (X_p Y_{2j} - X_{2j}Y_p Z_{2j})) \end{aligned} \tag{4.33}$$

Avec le fait que :

$$R = R_{2n-1}\xi^{n-1}\sigma + R_{2n-2}\xi^{n-2}\sigma + ... + R_n\sigma + R_{n-1}\xi^{n-1} + R_1\xi + R_0 \tag{4.34}$$

$$x = x_0 + x_1\xi + ... + x_{n-1}\xi^{n-1} \tag{4.35}$$

$$y = y_0 + y_1\xi + ... + y_{n-1}\xi^{n-1} \tag{4.36}$$

Par les cinq équations 4.32, 4.33, 4.34, 4.35 et 4.36 on peut tirer le système :

$$\begin{cases} f_0(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = a_0, \\ f_1(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = a_1, \\ . \\ . \\ f_{2n-2}(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = a_{2n-2}, \\ f_{2n-1}(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = a_{2n-1}, \\ {Y_P}^2 - {X_P}^3 + 3X_P - b = 0, \\ {Y_j}^2 - {X_j}^3 + 3X_j{Z_j}^4 - b{Z_j}^6 = 0, \\ {Y_{2j}}^2 - {X_{2j}}^3 + 3X_{2j}{Z_{2j}}^4 - b{Z_{2j}}^6 = 0, \\ X_{2j} = -8X_j{Y_j}^2 + 9({X_j}^2 - {Z_j}^4)^2, \\ Y_{2j} = 3({X_j}^2 - {Z_j}^4)(4X_j{Y_j}^2 - X_{2j}) - 8{Y_j}^4, \\ X_{2j} = 2Y_jZ_j, \end{cases} \tag{4.37}$$

Après avoir appliqué la mutation proposé dans l'algorithme de Miller modifié, le système 4.37 va changer à :

$$\begin{cases} f_{0_{r_2}}(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = b_0, \\ f_{1_{r_2}}(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = b_1, \\ . \\ . \\ f_{2n-2_{r_2}}(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = b_{2n-2}, \\ f_{2n-1_{r_2}}(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = b_{2n-1}, \\ f_{2n} = {Y_P}^2 - {X_P}^3 + 3X_P - b = 0, \\ f_{2n+1} = {Y_j}^2 - {X_j}^3 + 3X_j{Z_j}^4 - b{Z_j}^6 = 0, \\ f_{2n+2} = {Y_{2j}}^2 - {X_{2j}}^3 + 3X_{2j}{Z_{2j}}^4 - b{Z_{2j}}^6 = 0, \\ f_{2n+3} = X_{2j} + 8X_j{Y_j}^2 - 9({X_j}^2 - {Z_j}^4)^2 = 0, \\ f_{2n+4} = Y_{2j} - 3({X_j}^2 - {Z_j}^4)(4X_j{Y_j}^2 - X_{2j}) + 8{Y_j}^4 = 0, \\ f_{2n+5} = X_{2j} - 2Y_jZ_j = 0, \end{cases} \tag{4.38}$$

La résolution de ce système (multi-variants) repose sur la recherche des bases de Gröbner 42 qui l'engendre et sur la méthode d'élimination des termes.

Soit $<$ un ordre monomial défini sur l'ensemble des monômes de $F_p[X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}]$.
La recherche d'une base de Gröbner se fait en général par le calcul du S-polynôme :
Étant donné f_1 et f_2

$$S(f_1, f_2) = u_1 f_1 - u_2 f_2 \tag{4.39}$$

où lcm = LM(f_1) $\bigvee$ $LM(f_2)$ et $u_i = \frac{lcm}{LT(f_i)}$ pour i = {1, 2}.
Avec le fait que :
Soit i $\in$ {1, 2}.

— $LM(f_i)$ représente le monôme de tête de f_i, il est défini par :
$LM(f_i)$ = X^ρ, où ρ = max{$\alpha \in N^n$ tel que : les coéfficients de $f_i \neq 0$ }.

— LC(f_i) est le coéficient dominant de f_i, il est défini par LC(f_i) = coéfficient(f_i)$_\rho$.

— LT(f_i) est le terme de tête de f_i, il est défini par LT(f_i) = LC(f_i) . LM(f_i)

Le système 4.38 ne peut pas être résolu suite au résonnement suivant :

Résonnement

Premièrement, le r_2 figure forcement dans le calcul des S-polynôme de f_{ir_2} pour tout i $\in$ {0, 1, ..., 2n-1}, et ceci quelques soit la fonction pris dans le système 4.38, en effet :
Comme on a multiplié chaque fonction de 4.38 (celles de {0, ..., 2n-1}) par r_2, en plus, $r_2 \in F_p$, alors, le r_2 préexiste dans tout les coéfficients dominants des f_{ir_2} et ceci pour tout i $\in$ {0, 1, ..., 2n-1}.
On a, $\forall$ (i,j) $\in \{0, 1, ..., 2n-1\}^2$:

$$\begin{aligned} S(f_{ir_2}, f_{jr_2}) &= \frac{lcm}{r_2 LC_{reste}(f_{ir_2})} f_{ir_2} - \frac{lcm}{r_2 LC_{reste}(f_{jr_2})} f_{jr_2} \\ &= r_2'\left(\frac{lcm}{LC_{reste}(f_{ir_2})} f_{ir_2} - \frac{lcm}{LC_{reste}(f_{jr_2})} f_{jr_2}\right) \end{aligned} \tag{4.40}$$

Et que, $\forall$ (i,j) $\in$ {0, 1, ..., 2n-1} $\times$ {2n, ..., 2n+5} :

$$\begin{aligned} S(f_{ir_2}, f_j) &= \frac{lcm}{r_2 LC_{reste}(f_{ir_2})} f_{ir_2} - \frac{lcm}{LC(f_j)} f_j \\ &= r_2'\left(\frac{lcm}{LC_{reste}(f_{ir_2})} f_{ir_2} - r_2 \frac{lcm}{LC(f_j)} f_j\right) \end{aligned} \tag{4.41}$$

Pour trouver une base de Gröbner convenable au système 4.38, il suffit d'utiliser l'algorithme de Buchberger 42 :

Algorithme 4.5.2 : Buchberger

Entrée : I = $< f_{0_{r_2}}, ..., f_{2n-1_{r_2}}, f_{2n}, ..., f_{2n+5} > \in F_p[X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}]$

Sortie : G base de Gröbner de I.

1. G $\longleftarrow$ $f_{0_{r_2}}, ..., f_{2n-1_{r_2}}, f_{2n}, ..., f_{2n+5}$
2. CA $\longleftarrow$ {S(f_i, f_j), 0 $\leq i, j \leq$ 2n+5}
3. **Tant que** CA $\neq$ 0 **faire**
4. Choisir s $\in$ CA et l'extraire de CA
5. r $\longleftarrow$ s div G
6. **Si** r $\neq$ 0 **alors**
7. CA $\longleftarrow$ CA $\bigsqcup$ {S(g,r), g $\in$ G}
8. G $\longleftarrow$ G $\bigsqcup$ {r}
9. **Fin Si**
10. **Fin tant**

Retourner G

Le calcul de l'étape 5 se fait par : r $\longleftarrow$ r + LT(S(f_i, f_j)) (division de deux polynômes à plusieurs variables). En se basant sur les deux équations 4.40 et 4.41, la base de Gröbner du système 4.38 est alors :

$$\begin{cases} g_{0_{r_2}}(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = 0, \\ g_{1_{r_2}}(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = 0, \\ . \\ . \\ . \\ g_{2n-2_{r_2}}(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = 0, \\ g_{2n-1_{r_2}}(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = 0, \\ g_{2n_{r_2}}(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = 0, \\ g_{2n+1_{r_2}}(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = 0, \\ g_{2n+2_{r_2}}(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = 0, \\ g_{2n+3_{r_2}}(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = 0, \\ g_{2n+4_{r_2}}(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = 0, \\ g_{2n+5_{r_2}}(X_P, Y_P, X_j, Y_j, Z_j, X_{2j}, Y_{2j}, Z_{2j}) = 0, \end{cases} \quad (4.42)$$

La deuxième étape de la résolution du système 4.42 consiste de faire des éliminations de telle manière à trouver un polynôme d'un seul variable ; mais dans notre cas, le polynôme trouvé sera de deux variable, suite a l'existence de r_2 (ou l'inverse de r_2 qui est r_2'). Par conséquent, le système 4.38 ne peut pas être résolu.

4.6 Conclusion et perspectives

Nous avons vu dans ce chapitre une méthode qui résout le problème de Key Escrow pour le chiffrement basé sur l'identité, notre approche est uniquement basée sur le protocole de Diffie Hellman. Contrairement aux anciennes méthodes où il faut être prudent et suivre la syntaxe des cryptosystèmes demandés et la modifier selon le besoin, notre clé résultante peut être intégrée dans n'importe quel crypto système sans le changer. Les premières propositions fournies tablent sur une clé publique générée par l'utilisateur ce qui n'est pas toujours sûr puisqu'elle peut être remplacée et donc tomber facilement à l'attaque DoD. Nous remarquons que l'idée de Joseph K. Liu et al [129] ne suffit pas pour défendre le problème DoD, comme elle nécessite en cas d'une vérification non valide de demander au récepteur d'envoyer la clé publique nécessaire, le problème préexiste encore puisque cette dernière n'est pas certifiée. Dans notre méthode où on utilise la clé de Diffie-Hellman, nous vérifions les clés partielles et nous sommes encadrés par la PKG, en cas des vérifications non valides nous contactons la PKG et non pas l'utilisateur pour régler les choses.
En plus, la performance de notre approche se voit dans le cadre de la complexité. Nous l'avons comparé avec le schéma de Yinxia Sun et al [166] tracé sans couplage, nous avons remarqué que tant que la communication s'élargie (tours plus grands), notre idée sera la plus efficace.
Aussi, nous avons comparé notre approche intégrée dans le schéma de Boneh-Franklin, avec le schéma de Aniket Kate et Ian Goldberg [117] (qui est aussi tracé avec Boneh Franklin). Ce dernier a pour but de conserver la clé maîtresse en la distribuant entre plusieurs autorités, ce qui résiste aussi au Key Escrow. Nous remarquons que si nous conservons bien la clé maîtresse (en fait on n'en pas besoin puisqu'on a une clé qui la cache), c'est mieux d'utiliser notre approche que le schéma de Aniket Kate et Ian Goldberg pour résister au Key Escrow.
Par ailleurs, nous venons de présenter la rigidité du chiffrement basé sur l'identité contre les attaques physiques. Nous avons exposé dans la partie 4.4.2 notre idée pour réussir l'attaque DPA contre le couplage. Ensuite, nous avons traduit cette idée sur la cryptographie IBC. Enfin, nous avons proposé les méthodes de résistance à l'attaque DPA en examinant les anciennes et en proposant d'autres méthodes. Nous avons aussi examiné les études faites contre l'attaque DFA et proposé une méthode pour défendre l'attaque dans le cas d'un degré de plongement pair.

Perspectives

Le chapitre qu'on a traité représente un complément dans notre thèse, il porte les perspectives suivantes :

- La première perspective marquée est de ne pas démontrer la résistance de notre approche qui concerne le Key Escrow contre les attaques de simulations, nous laisserons cela comme limite de cette thèse est non du protocole.
- Ainsi, nous avons formalisé l'attaque DPA seulement en théorie, nous viserons sa traduction en réalité.
- Dans le cadre DFA, nous espérons pouvoir proposer des méthodes plus convenables à la réussite d'un DFA.

Chapitre 5

Sécurité et réalisation d'IBE

Le présent chapitre consiste, tout d'abord, à fixer quelques conditions qui concernent la sécurité d'IBE avant d'exploiter les résultats dégagés dans les chapitres 2 et 3 .
La sécurité d'IBE est intimement liée aux problèmes de Diffie Hellman utilisés dans le schéma cible, sans oublier que CDHP et BDHP sont souvent les problèmes rencontrés au niveau de la preuve de la sécurité des schémas d'IBE. Dans ce cadre, on exploite une simple méthode menaçante de CDHP et BDHP et on s'est limité à l'utilisation du couplage de Tate. Ainsi et comme avec Frey et Rück, on répéte le nuisible rôle de Tate (et ses variantes) dans la cryptanalyse. Pour ce faire, nous citons dans les parties 5.1 et 5.2 les notions qui construisent le couplage : diviseurs, fonctions associées, fonctionnement de Miller etc et nous développons notre idée qui permet d'attaquer CDHP et BDHP dans la partie 5.4 en soulevant la fragilité du couplage de Tate (et ses variantes) par rapport à celui de Weil.
Avant d'exploiter les résultats tirés des chapitres 2 et 3 nous mettons l'accent sur l'implémentation d'un IBE. Pour ce faire, on détaille notamment la stratégie pour réussir cette implémentation, ensuite, on traite l'exemple du schéma basique du quatrième schéma pour l'IBE exposé dans la section 3.3 du chapitre 3 (travail sous le *report 2012/321*) en se basant sur le logiciel *Magma*, une deuxième implémentation sera citée sans détails dans la conclusion de ce chapitre en prenant en considération les conditions du *report 2012/277*.
Après avoir obtenu les résultats des implémentations, on évaluera certaines comparaisons, qui contiennent entre autre la comparaison entre les courbes adaptées au calcul d'un couplage. D'autres comparaisons concernent, les schémas avec couplage, les schémas sans couplage et ceux avec couplage.

5.1 Corps construisant le couplage

5.1.1 Genres des courbes elliptiques

Il y a deux genres de courbes elliptiques (voir chapitre 1 pour la définition), courbes supersingulières et courbes ordinaires [164]. Les deux sont différenciées selon la définition suivante :

Définition 5.1.1 : *Soit E une courbe elliptique définie sur un corps simple F_p (où composé F_q, avec q est sous forme d'une puissance).*
*On dit que E est **supersingulière**, si elle satisfait l'équivanlence entre les trois conditions suivantes :*

1. $\sharp E(F_p) \equiv 1 \bmod p$, *ou* $\sharp E(F_p) = p + 1 - t$, *avec* $p|t$ *(t est la trace)*
2. *E a un point non trivial d'ordre p dans* $\bar{F}_q$, *càd,* $E[p] = \infty$.
3. *L'endomorphism de l'anneau E dans* F_q *est non commutatif, ou plus précisément, il a un ordre dans l'algèbre de quaternion.*

Une courbe non-supersingulière est appelée ***Ordinaire***.

Ordre d'une courbe E (K), où K est un corps fini sous forme simple ou composée

L'ordre d'une courbe est donné par le célèbre théorème de Hasse :

Théorème 5.1.1 : *Soit E une courbe elliptique définie sur un corps K, alors :*

$$p + 1 - 2\sqrt[2]{p} \leq |E(F_p)| \leq p + 1 + 2\sqrt[2]{p} \tag{5.1}$$

Cette formule peut être simplifiée après avoir utilisé la trace de **Frobenius**.

Définitions 5.1.2 : *Le Frobenius, noté π_p, est une application de la courbe elliptique E(K) d'équation $y^2 = x^3 + ax + b$ sur la courbe elliptique notée $E^{(p)}(K)$ et d'équation $y^2 = x^3 + a^p x + b^p$. Il est donné par :*

$$\begin{aligned} \pi_p : E(K) &\longrightarrow E^{(p)}(K) \\ P = (x_P, y_P) &\longrightarrow \pi_p(P) = (x_P^p, y_P^p) \end{aligned} \tag{5.2}$$

Le Frobenius est un morphism qui vérifie :

$$\pi_p(P_1 + P_2) = \pi_p(P_1) + \pi_p(P_2) \tag{5.3}$$

Le polynôme caractéristique du Frobenius est l'endomorphism qui est nul sur la courbe elliptique. Il est défini par :

$$\pi_p(X) = X^2 - tX + det(\pi_p) \tag{5.4}$$

Où t représente la trace de π_p et det(π_p) son déterminant.

Théorème 5.1.2 : *Soit E une courbe elliptique sur K et t la trace du Frobenius sur E. Le cardinal d'une courbe elliptique E(F_p) où F_p est un corps simple, p est la caractéristique, est donné par :*

$$|E(F_p)| = p + 1 - t \tag{5.5}$$

En général, on peut utiliser l'algorithme de Schoof [158] publié en 1985 pour déterminer l'ordre d'un groupe, ou ses améliorations comme SEA (Atkin, Elkies et Couveignes).

Proposition 5.1.1 : *Pour* $|E(F_p)|$*=p+1-t, si* α *et* β *sont les racines complexes de* $X^2 - Tr(\phi_q)X + q$*, alors :*

$$|E(F_{p^k})| = p^k + 1 - tr({\phi_p}^k) = p^k + 1 - (\alpha^k + \beta^k) \tag{5.6}$$

Il est intéressant de remarquer que $tr_k := tr({\phi_p}^k) = \alpha^k + \beta^k$ vérifie la relation de récurrence :

$$tr_{k+2} = tr_1 tr_{k+1} - ptr_k \quad où \quad tr_1 = p + 1 - (|E(F_p)|). \tag{5.7}$$

5.1.2 Point de r-torsion

L'ordre d'un point P dans une courbe elliptique E est le plus petit entier m qui vérifie : mP=O, si ce paramètre n'existe pas on dit que P admet un ordre infini.
On dit que P est un point de r-torsion (r est un entier positive) si r P = O (point à l'infini).
Le point rP est à l'infini si l'abscisse de rP est nulle. Dans le cas où ces abscisses sont regroupés dans un corps fini, trouver les racines dont lesquelles une abscisse sera nulle n'est pas possible. C'est pourquoi il est conseillé de travailler dans des corps algébriquement clos.

Définition 5.1.3 : *Soit r un entier positif, l'ensemble E[r] =* $\{ P \in E(\overline{K})/rP = O\}$ *est appelé l'ensemble de point de r-torsion.*

$\overline{K}$ est le corps algébriquement clos. L'entier r n'est pas nécessairement l'ordre du point P il peut être son multiple.
Le corps $\overline{K}$ paraît plus large, pour esquiver ce problème il vaut mieux penser au corps étendu de K.

5.1.3 Degré de plongement d'une courbe elliptique

Définition 5.1.4 : *Soit E(F_p) une courbe elliptique et r un diviseur premier de $|E(F_p)|$. Le degré de plongement de la courbe elliptique E relativement à r est le plus petit entier k tel que r divise* $(p^k - 1)$.

Théorème 5.1.3 : *Soit E(F_p) une courbe elliptique définie sur un corps fini F_p et r un diviseur de $|E(F_p)|$. Si r est premier avec p et r ne divise pas (p - 1), alors E[r] $\subset E(F_{p^l})$ pour l un entier positif, si et seulement si, r divise* $(p^l - 1)$.

D'après ce théorème, le corps étendu qu'on cherche n'est autre que F_{p^k} où k est le degré de plongement.

5.1.4 Distorsions

Les distorsions sont initialisées par Verheul [172]. Elles sont utilisées pour munir des isomorphisms entre les courbes elliptiques sur F^2 qui ont l'ordre $p^2 - p + 1$. Les distorsions ont l'avantage d'envoyer les points sous une forme indépendante. Cependant, trouver des élements indépendants est un point important pour les couplages.

Exemple de distorsion

Corps	Courbe	Distorsion
F_p	$y^2 = x^3 + ax$	$\phi(x, y) = (-x, iy)$
F_p	$y^2 = x^3 + b$	$\phi(x, y)$=(ζ x,y)

TABLE 5.1 – Quelques distorsion

Propriété 5.1.1 (Verheul)[132] :

Soit E /F_{p^k} une courbe supersingulière et soit P ∈E(F_{p^k}) [n].
Si n est relativement premier avec la caractéristique p de F_{p^k}. Alors il existe une distorsion relativement liée à p.

Propriété 5.1.2 (Verheul)[132] :

Soit E(F_{p^k}) une courbe elliptique ordinaire et soit P ∈ E(F_{p^k}) [n].
Si n est relativement premier avec la caractéristique p de F_{p^k} et E[n] n'appartient pas à E(F_{p^k}), alors il n'existe aucune distorsion relativement liée à p.

Les courbes supersingulières peuvent alors bénéficier des distorsions, tandis que les courbes ordinaires ne le sont pas. Mais en général, pour une courbe supersingulière ou ordinaire, on doit connaître si elle a une forme égale à celle de la courbe tordue. On peut alors utiliser un difféomorphisme lié à cette dernière.

5.1.5 Quelques préliminaires sur les courbes tordues

Définition 5.1.5 : *Soient E et $\widetilde{E}$ deux courbes elliptiques définies sur F_p. On dit que la courbe $\widetilde{E}$ est une tordue (ou twist) de degré d de la courbe E, s'il existe un isomorphism φ_d défini sur F_{p^d} de $\widetilde{E}$ dans E, tel que d soit minimal. Avec :*

$$\varphi_d : \widetilde{E}(F_p) \longrightarrow E(F_{p^d}) \tag{5.8}$$

Théorème 5.1.4 [164] : *Soit E une courbe elliptique d'équation sous forme de Weirstrass : $y^2 = x^3 + ax + b$, définie sur une extension F_{p^k}. Suivant les valeurs de k, les degrés d possibles des twists sont : 2, 3, 4 et 6.*

• Twist de degré quadratique

Soit $E(F_p) : y^2 = x^3 + ax + b$ une courbe elliptique, v un point de F_p^* et qui n'est pas une racine dans F_p^*. Alors :

$$\widetilde{E}/F_p : y^2 = x^3 + v^2ax + v^3b \tag{5.9}$$

est appelé une twist quadratique de E. $\widetilde{E}$ est isomorphe à E dans une extension de degré 2 dans F_p.

• Twist d'un degré grand

Pour toute courbe E(F_p), il est possible d'obtenir différente twist à partir de celles qui sont quadratiques. Dans ce cas on a :

$$\widetilde{E} : y^2 = x^3 + a'x + b' \quad avec \quad a' = v^{\frac{4}{d}}a \quad et \quad b' = v^{\frac{6}{d}}b \tag{5.10}$$

v est la racine de degré exactement d (non moins) dans F_p. Toutes ces twists sont isomorphs à E (F_{p^d})

Les twists (tordues) possibles (qui soit quadratique ou d'un degré grand) sont regroupés dans le tableau 5.2. Mais dans tous les cas : p = 1(mod d) est une condition nécessaire. Nous regroupons aussi dans le tableau les isomorphisms $\varphi_d : \widetilde{E} \mapsto$ E, avec d est le degré d'un twist.

d	P ∈ E	P ∈ $\widetilde{E}$	E	$\widetilde{E}$	φ_d
2	(x,y)	$(vx, v^{\frac{3}{2}}y)$	$y^2 = x^3 + ax + b$	$y^2 = x^3 + v^2ax + v^3b$	$(v^{-1}x, v^{-\frac{3}{2}}y)$
3	(x,y)	$(v^{\frac{1}{3}}x, v^{\frac{1}{2}}y)$	$y^2 = x^3 + b$	$y^2 = x^3 + vb$	$(v^{-\frac{1}{3}}x, v^{-\frac{1}{2}}y)$
4	(x,y)	$(v^{\frac{1}{2}}x, v^{\frac{3}{4}}y)$	$y^2 = x^3 + ax$	$y^2 = x^3 + vax$	$(v^{-\frac{1}{2}}x, v^{-\frac{3}{4}}y)$
6	(x,y)	$(v^{\frac{1}{3}}x, v^{\frac{1}{2}}y)$	$y^2 = x^3 + b$	$y^2 = x^3 + vb$	$(v^{-\frac{1}{3}}x, v^{-\frac{1}{2}}y)$

TABLE 5.2 – Types possibles des twists

Dans le taleau 5.2, à chaque twist correspond l'isomorphism convenable. Selon d, le paramètre v vérifie $x^d - v$ est irréductible sur F_{p^d}.
L'objectif premier de la construction d'un twist est qu'au lieu de regrouper les calculs dans F_{p^k}, nous les regroupons seulement dans F_{p^d} où d divise k.

5.1.6 Fonction rationelle

Soit E(k) une courbe elliptique d'équation f(X,Y) = 0. La fonction g est un zéro de E(k), si g est un multiple de f. On peut définir une relation d'équivalence sur l'anneau k[E] comme étant : k[X,Y]/(f). Le corps k(E) est un corps de fonction rationnelle sur E.
Une fonction est rationnelle sur k(X, Y), si elle envoie au moins un point de E(k) sous une forme rationnelle. Par exemple, E(F_{q^k}) peut s'écrire comme étant $\frac{g(X,Y)}{h(X,Y)}$ avec $g(X,Y)$ et $h(X,Y)$ sont deux fonctions sur $E[F_{q^k}]$ où par exemple, $g(X,Y)$ peut être écrit sous la forme :

$$g(X,Y) = g_x(X) + Y g_y(Y) \tag{5.11}$$

Avec g_x,$g_y \in F_{q^k}[X]$, après avoir utilisé $Y^2 = X^3 + aX + b$ (équation d'un corps de caractéristique différente de 2 et 3)
Pour une fonction rationnelle :

$$\frac{g(X,Y)}{h(X,Y)} \tag{5.12}$$

Les zéros de $g(X,Y)$ sont appelés les zéros de la fonction rationnelle, et les zéros de $h(X,Y)$ sont ses pôles.
Les fonctions rationnelles qui sont définies sur une courbe elliptique E, admettent un nombre de zéros égal aux nombres de pôles (pour plus de détails voir [74]).

5.1.7 Diviseurs

Un diviseur est un concept universel qui relie les zéros et les pôles avec leurs ordres de multiplicités. On peut écrire un diviseur D comme étant :

$$D = a_1[P_1] + a_2[P_2] + a_3[P_3] + a_4[P_4] + a_5[P_5] \tag{5.13}$$

Avec $a_1, a_2, a_3, a_4, a_5 \in$ Z sont les zéros ou les pôles des points P_1, P_2, P_3, P_4, P_5 d'une courbe elliptique E.
Prenons l'exemple : $D = [2P] - 3[Q] - 2[P]$. Dans ce diviseur [2P] et 2[P] sont différents. Le premier point [2P] a un zéro d'ordre 1, tandis que le deuxième 2[P] à un zéro d'ordre -2.

Un diviseur est alors un ensemble généré par le symbole [P]. L'ensembles des diviseurs est noté par Div(E), c'est le groupe abélien libre engendré par les points de la courbe, noté par :

$$\{Div_k(E) = \sum_{P \in E} n_p(P) : n_P \in Z \ \ presque \ \ tous \ \ nuls\} \tag{5.14}$$

Si D $\in$ Div(k), l'ensemble des points P $\in$ E(k) tels que $n_P \neq 0$ est appelé support de D et noté supp(D).

Le degré de D est un entier qui a la valeur suivante :

$$deg(\sum_i a_i[P_i]) = \sum a_i \tag{5.15}$$

Sa somme est :

$$sum(\sum_i a_i[P_i]) = \sum a_i P_i \tag{5.16}$$

On note l'ensemble des diviseurs de degré 0 par Div^0

On dit qu'un diviseur D=$\sum_i a_i P_i$ est principal, s'il a un degré 0 et si $sum(\sum_i a_i[P_i]) = O$. L'ensemble des diviseurs principaux est noté par Prin(E).

Deux diviseurs D et D' sont équivalents si leur différence est un élément dans Prin(E).

Un diviseur de fonction rationnelle est défini comme suit :

$$div(f) = \sum_{p \in E(k)} ord_p(f)[P] \tag{5.17}$$

Avec $ord_P(f)$ est l'ordre de f dans le point P (zéros ou pôles). En plus, d'après [74] chaque fonction rationnelle à un degré 0.

Pour un diviseur D=$\sum_i a_i P_i$ et une fonction rationnelle f on a : f(D)=$\prod f(P_i)^{a_i}$, il faut noter que le support de D et div(f) doivent être disjoints.

Exemples de diviseurs pour des fonctions rationnelles connues

Ici, on essaye de trouver le diviseur d'une fonction qui a comme fonction rationnelle : ax+by+c (équation d'une ligne qui passe par les points P_1 et P_2 d'une courbe elliptique E, où $P_1 \neq \pm P_2$). Cette ligne coupe la courbe dans un troisième point P_3. Alors la fonction f(x,y)=$ax + by + c$ à trois zéros dans les points P_1, P_2 et P_3 ; un pôle d'ordre 3 dans le point O. En effet, f à trois zéros et puisqu'elle est une fonction rationnelle (son degré est donc égal à 0), elle a nécessairement un pôle d'ordre 3 dans O (point à l'infini). Alors :

$$div(f) = P_1 + P_2 + P_3 - 3O \quad càd \quad (ax + by + c) = P_1 + P_2 + P_3 - 3O \tag{5.18}$$

Pour l'équation d'une ligne verticale $x - x_Q = 0$ (ligne qui passe par Q et -Q). De même que précédemment on trouve :

$$div(x - x_Q) = [Q] + [-Q] - 2O \tag{5.19}$$

5.2 Un peu de théorie sur les couplages

Un couplage est une application bilinéaire qui prend deux points d'une courbe elliptique et fait sortir un élément du groupe multiplicatif de $r^{ième}$ racine de l'unité. Les premiers introduits en cryptographie sont les couplages de Weil et de Tate. Plus récemment, le couplage de Tate a connu des améliorations conduisant à ce qu'on appelle Optimal Pairing [171][101] (Ate, Twisted Ate, Eta..).

5.2.1 Formules explicites des couplages

Avant de donner les formules explicites de quelques couplages, nous démontrons tout d'abord les équivalences suivantes :
$[P] - [O]$ et $[P + R] - [R]$ sont équivalents quelque soit le point R de la courbe elliptique mise en point. En effet, selon ce qu'on a annoncé précédemment, on a :

$$div(u) = [P] + [R] + [-(P + R)] - 3[O] \quad et \quad div(v) = [-(P + R)] + [P + R] - 2[O] \tag{5.20}$$

Avec u est la ligne qui passe par les points : P, R, -(P+R) et v est la ligne qui passe par les points P+R et -(P+R). Alors :

$$[P] - [O] = [P + R] - [R] + div(\frac{u}{v}) \tag{5.21}$$

Par conséquent, on peut écrire le diviseur D_P par l'une des expressions : $[P]-[O]$ ou $[P+R]-[R]$.

Comme nous avons donné dans le chapitre 1 la forme des deux couplages Tate et Ate, nous donnons ici celle de Weil

Couplage de Weil

Soient : P, Q $\in E[r]$ et D_P, D_Q deux diviseurs de degré 0 et de supports différents. Alors on peut prendre D_P équivalent à $[P] - [O]$ et D_Q équivalent $\grave{a}$ [Q + S]-[S], avec S est choisi de manière que P, O, Q+S, S sont différents l'un de l'autre.

Comme P et Q ∈ E[r], alors rD_P et rD_Q sont des diviseurs principaux. Donc il existe deux fonctions principales f_P et f_Q telles que :

$$divf_P = rD_P \quad et \quad divf_Q = rD_Q \tag{5.22}$$

Le couplage de Weil est alors défini comme suit :

$$\begin{aligned} e_{wr} : E(F_p)[r] \times E(F_{p^k})[r] &\to \mu_r \\ (P,Q) \ &\to e_{wr}(P,Q) = \frac{f_{D_Q}(D_P)}{f_{D_P}(D_Q)} \end{aligned} \tag{5.23}$$

Le μ_r est l'ensemble des $r^{ième}$ racines de l'unité.

Dans l'expression des couplages figure la fonction rationnelle f_r, pour la calculer on fait appel habituellement à l'algorithme de Miller.

Façade de l'algorithme de Miller : Fonctionnement itératif

L'algorithme de Miller calcule $f_{D_P}(Q)$ en utilisant la méthode suivante :
Pour un diviseur $D_P = [P + R] - [R]$ et un entier positif i, nous définissons les diviseurs suivants :

$$D_i = i[P + R] - i[R] - [iP] + [O] \tag{5.24}$$

Puisque ce diviseur est principal, alors il existe une fonction rationnelle f_i telle que $div(f_i) = D_i$. Pour i=r on a :

$$D_r = r[P + R] - r[R] - [rP] + [O] = rD_P \tag{5.25}$$

Du fait que : rP=O alors $f_r = f_{D_P}$, le problème se focalise donc au calcul de $f_r(Q)$.
Étant donné : $f_{r_1}(Q)$ et $f_{r_2}(Q)$ ou r_1 et r_2 sont deux entiers positives, r_1P, r_2P et $(r_1+r_2)P$ trois points de la courbe elliptique E. On veut calculer $f_{r_1+r_2}$.
Par définition, on a :

$$D_{r_1} = r_1[P + R] - r_1[R] - [r_1P] + [O] \ \ et \ \ D_{r_2} = r_2[P + R] - r_2[R] - [r_2P] + [O], \tag{5.26}$$

$$D_{r_1+r_2} = (r_1 + r_2)[P + R] - (r_1 + r_2)[R] - [(r_1 + r_2)P] + [O] \tag{5.27}$$

Nous définissons l'équation : $aX+bY+c = 0$ comme étant la ligne qui passe par les deux points r_1P et r_2P (si $r_1 = r_2$ on obtient l'équation de la tangente), posons $L_{r_1P,r_2P}(X, Y) = aX + bY + c$. Soit encore $X + d = 0$ la ligne verticale qui passe par le point $(r_1 + r_2)P$, définissons la fonction $V_{(r_1+r_2)}(X, Y) = X + d$.

En utilisant ce qu'on a annoncé précédemment, on a :

$$div(L_{r_1P,r_2P}) = [r_1P] + [r_2P] + [-(r_1 + r_2)P] - 3O \tag{5.28}$$

$$div(V_{(r_1+r_2)}) = [(r_1 + r_2)P] + [-(r_1 + r_2)P] - 2O \tag{5.29}$$

Alors :

$$D_{r_1+r_2} = D_{r_1} + D_{r_2} + div(L_{r_1P,r_2P}) - div(V_{(r_1+r_2)}) \tag{5.30}$$

Ce qui implique que :

$$f_{(r_1+r_2)}(Q) = f_{(r_1)}(D_Q) \cdot f_{(r_2)}(D_Q) \cdot \frac{L_{r_1P,r_2P}Q}{V_{(r_1+r_2)}Q} \tag{5.31}$$

Puisqu'on a :

$$divf_{r_1+r_2}(Q) = divf_{r_1}(Q) + divf_{r_2}(Q) + div\frac{L_{r_1P,r_2P}}{V_{(r_1+r_2)}} \tag{5.32}$$

Et donc :

$$\sigma(f_{(r_1)}(Q), f_{(r_1)}(Q), r_1P, r_2P, (r_1 + r_2)P) = f_{(r_1+r_2)}(Q) \tag{5.33}$$

σ représente la sortie de l'algorithme de Miller.

Premières allures de l'algorithme de Miller

Soit k un entier, on a vu que l'algorithme de Miller suit une manière itérative qui fait sortir la fonction rationnelle f_k.

L'algorithme de Miller se base sur les trois choses suivantes :

— L'expression (dans la base binaire) de l'entier r qui est l'ordre du premier point, où $r = \sum_{i=0}^{i=m}(r_i 2^i)$

— $(f_0) = 1$

— $(f_1) = (\frac{V_{P+R}}{L_{P,R}})$.

Le fait que : $(f_0) = 1$ est clair. Pour (f_1), on a :

$$div(aX+bY+c) = [P]+[Q]+[R]-3O \ \ (aX+bY+c \ est \ la \ ligne \ qui \ passe \ par \ P, Q, R) \tag{5.34}$$

et

$$div(X - X_Q) = [Q] + [-Q] - 2O. \tag{5.35}$$

Alors :

$$\frac{div(aX + bY + c)}{div(X - X_Q)} = [P] + [R] - [-Q] - [O] \tag{5.36}$$

Mais, $P + R = -Q$, donc

$$[P + R] - [P] - [R] + [O] = (\frac{V_{P+R}}{L_{P,R}}) = (f_1) \tag{5.37}$$

Ce qui prouve la relation et la récurrence pour i = 1.

Miller sous forme algorithmique :

Suivant ces façons itératives, Miller [137] a exploité un algorithme qui est bien utile pour calculer les fonctions rationnelles figurantes dans le calcul des couplages. Nous donnons la forme améliorée de cet algorithme, càd celle où on utilise moins d'inverses.

***Algorithme 5.2.1 : Miller(P, Q, r)* (amélioré sous coordonnées affines)**

Entrée : r = (r_n**...**r_0**)(représentation binaire)**, P = $(x_P, y_P) \in E[r](\subset$ E(F_p)) et

Q = $(x_Q, y_Q) \in G_1(\subset E(F_{p^k}))$

Où E est une courbe elliptique d'équation sous la forme :

$y^2 = x^3 + ax + b$

Sortie : $f_{r,P}(Q) \in G_3$ ($\subset F^*_{p^k}$)

T=$(x_T, y_T) \leftarrow$ P

$f \leftarrow 1$

$g \leftarrow 1$

Pour i = n - 1 à 0 **faire**

1. $f \leftarrow f^2 \times (y_Q - y_T - \lambda x_Q + \lambda x_T)$

   $g \leftarrow g^2 \times (x_Q - x_{2T})$

   Où $\lambda = \frac{3x_T + a}{2y_T}$

   T $\leftarrow (\lambda^2 - 2x_T, \lambda(x_T - \lambda^2 + 2x_T) - y_T)$

2. **Si** r_i=1 **alors**

   $f \leftarrow f \times (y_Q - y_P - \lambda x_Q + \lambda x_P)$

   $g \leftarrow g \times (x_Q - x_{P+T})$

   Où $\lambda = \frac{x_T - x_P}{y_T - y_P}$

   T $\leftarrow (\lambda^2 - x_T - x_P, \lambda(x_T - \lambda^2 + x_T + x_P) - y_T)$

Retourner $\frac{f}{g}$

5.3 Sécurité des IBE

La sécurité des schémas d'IBE basés sur les couplages repose sur les trois points suivants :

1. Sécurité des fonctions de hachage.
2. Celle de EDLP càd PDL sur les courbes elliptiques.
3. Sécurité du couplage.

La sécurité des fonctions de hachage a été bien étudiée dans la la littérature elle est dernièrement couronnée par la sélecton de Keccak pour la candidature SHA3, cette dernière est surement sécurisé. Mais pour appliquer les fonctions de hachage aux schémas des IBE sous Random Oracle, nous faisons soit : des concaténations, des réductions modulaires ou autres techniques. Effectivement, cela demande d'étudier la sécurité de ces fonctions selon ces changements. Malheureusement, nous remarquons que personne n'a pris l'initiative pour étudier ce genre de fonctions et nous avérons que la présente thèse n'apporte aucune solution à ce problème.

Quant à la sécurité du deuxième point, il concerne tous les schémas cryptographiques. Pour attaquer un EDLP, nous pouvons utiliser l'un des algorithmes suivants :

— Pollard Rho qui a une complexité exponentielle.

— Pohling Hellman, qui a une complexité polynômiale. Pour lui résister, il suffit de prendre un nombre premier.

— FFS (Function Field Sieve), qui a une complexité sous exponentielle, cet algorithme adresse uniquement la carctéristique 2 et 3 et les courbes supersingulières.

— NFS (Number Field Sieve), cet algorithme à une complexité sous exponentielle.

Nous renvoyons à la thèse [47] pour un panorama et une discussion à propos de ces quatres algorithmes.

Concernant le troisième point, les attaques les plus affrontées dans la littérature sont connues par MOV [134](utilisation du couplage de Weil) ou encore FR [78] (utilisation du couplage de Tate). Le principe de ces deux attaques est d'après ce qui suit :
Attaquer r à partir de rP, peut se faire après l'avoir attaqué lors de son utilisation dans le couplage. En effet, pour assurer la sécurité d'un couplage il faut tenir en compte le logarithme discret rP et aussi la sécurité de p^k où vive le résultat e(rP,Q). Le fait que e(rP,Q)=$e(P,Q)^r$ simplifie l'attaque de r. Les tailles minimales selon le NIST de r et p^k dans le couplage qui

permettent de travailler dans un niveau de sécurité équivalent à celui de la cryptographie symétrique (par exemple le protocole AES) sont donnés dans le tableau 5.3.

Niveau de sécurité (en bits)	80	128	192	256
Nombre minimal de bits de r	160	256	384	512
Nombre minimal de bits de p^k	1,024	3,072	7,680	15,360

TABLE 5.3 – Niveau de sécurité nécessaire pour r l'ordre du premier coordonnée du couplage et p^k

La sécurité de la cryptographie basée sur les couplages ne se base pas uniquement sur EDLP et PDL ; mais les problèmes tels que CDHP et BDHP et d'autres (problèmes bilinèaires de Diffie-Hellman) peuvent influencer sur sa sécurité.
Pour étudier la sécurité de CDHP et BDHP, certains chercheurs utilisent ce qu'on appelle pairing inversion.

5.3.1 Difficultés d'inverser les couplages (pairing inversion) selon Galbraith et al [85]

Selon [31], Verheul et Satoh [172][157] ont étudié la sécurité de CDHP et BDHP sous les couplages (méthodes connues dans la littérature par Pairing Inversion), ils déclarent [172][157] que si on peut trouver en même temps RPI et LPI càd attaquer GPI, on peut résoudre CDHP. Avec :

Définition 5.3.1 : *Right Pairing Inversion problem (**RPI**).*
Étant donné $P \in G_1$ et $\zeta \in G_T$, trouver $Q \in G_2$ qui satisfait e(P,Q) = ζ.

Définition 5.3.2 : *Left Pairing Inversion problem (**LPI**).*
Étant donné $Q \in G_2$ et $\zeta \in G_T$, trouver $P \in G_1$ qui satisfait e(P,Q) = ζ.

Définition 5.3.3 : General Pairing Inversion problem *(**GPI**).*
Étant donné $\zeta \in G_T$, trouver $(P,Q) \in G_1 \times G_2$ sachant que e(P,Q) = ζ.

La définition 5.3.3 généralise les deux premières. Si on réussit l'attaque de GRI, on peut aussi briser CDHP. Verheul [172] trouve que cette tâche n'est pas possible. Galbraith et al [85] ont ensuite étudié la performance et la possibilité des approches [172][157], selon les cas suivants :

- **Cas où les groupes sont cycliques :**

  En utilisant des homomorphismes, Galbraith et al [85] ont réduit le besoin de RPI et LPI à un seul. Leurs résultats se résument ainsi :

  Theorème 5.3.1 : *Soit e :* $G_1 \times G_2 \longrightarrow G_T$ *un couplage non-dégénéré, les* G_1, G_2 *et* G_T *des groupes d'ordre premier r. Les assertions suivantes sont équivalentes.*

  1. *Résoudre LPI et RPI dans un temps polynômial.*
  2. *Résoudre LPI dans un temps polynômial et tout homomorphism* $G_1 \longrightarrow G_2$ *peut être calculer dans un temps polynômial.*
  3. *Résoudre RPI dans un temps polynômial et tout homomorphism* $G_2 \longrightarrow G_1$ *peut être aussi calculer dans un temps polynômial.*

Si on trouve une de ces équivalences (voir [85] pour la preuve) dans un temps polynômial, on peut aussi résoudre le CDHP dans le même temps. Mais cela, n'est pas évident d'après [85].

- **Cas général :**

  Ça veut dire, cas où les groupes qui rentrent dans la construction ont une forme générale. Dans ce cas, les définitions de LPI, RPI et GPI sont les mêmes sauf que cette fois, nous prenons ζ dans μ_r et nous formalisons les définitions suivant des diviseurs (voir [85]). Ainsi, lors des démonstrations, Galbraith et al ont ajouté la définition de Miller Inverse (MI).

  Définition 5.3.4 : *Soit* D_1 *un diviseur fixé et soit S un ensemble des diviseurs. Soit z* $\in F_{q^k}$*, réussir un MI est d'après le fait de calculer un diviseur* $D_2 \in$ *S tel que z =* $f_{r,D_1}(D_2)$*. Si aucun diviseur n'existe, faire sortir 'aucune solution'.*

Les auteurs ont examiné l'inverse des couplages sous forme exponentiation (Tate et ses variantes), qui demande d'inverser l'exponentiation et le MI. Galbraith et al [85] ont remarqué que si on sait inverser MI (cas du couplage de Ate muni d'un paramètre petit), l'exponentiation nous bloque et si on gère le problème de l'exponentiation (prendre n'importe quelle racine [85]), le MI nous bloque. Dans tous les cas l'inverse des accouplements n'est pas possible.
Nous présentons par la suite une méthode où on peut attaquer CDHP et BDHP, si on ne prend pas quelques précautions.

5.4 Cryptanalyse de CDHP et BDHP, sous certaines conditions, le couplage de Tate est moins sécurisé que celui de Weil

Bien que, la recherche s'accumule dans le côté arithmétique des couplages et ainsi comparé Weil et Tate [58] [67] [75] du point de vue efficacité ; mais, il est temps aussi de tester le poids de chacun des deux au niveau sécurité. Il est usuel d'après la discussion citée au dessus de Galbraith et al [85] que le couplage de Weil est plus sécurisé que celui de Tate, puisque, pour attaquer CDHP, avec Weil, cela demande de réussir $M_F I$ (Inverse de Miller Full càd Miller initialisé par un point du corps composé F_{p^k}) et $M_L I$ (Inverse de Miller Lite càd Miller initialisé par un point du corps simple F_p). Tandis qu'avec Tate, nous aurons besoin de $M_L I$ et l'inverse d'une exponentiation. Galbraith et al [85] ont exposé une méthode où on peut se servir d'une manière générale de la racine $d^{ième}$ de l'exponentiation (d est l'exposant). Alors que le MI ne peut se faire que dans des cas très particuliers. Par conséquent, inverser Tate est plus facile qu'inverser Weil. Nous ajoutons à cette ramarque, deux autres avec lesquelles nous montrons que le couplage de Tate (et ses variantes) serve bien la cryptanalyse :

— L'attaque FR [78] est plus fragile que celle de MOV [134]

— Les propriétés de Tate (et ses variantes) peuvent conduire aux attaques de CDHP et BDHP.

Au sens de la dernière remarque, nous proposerons [9] dans les parties qui suivent une continuation des travaux de Verheul [172] et Satoh [157]. Mais, au contraire de leur stratégie et en utilisant le couplage de Tate, nous proposons une autre sorte d'attaque de CDHP et BDHP. Qui demande de passer par plusieurs étapes. Examinons tout d'abord la linéarité de certains difféomorphisms.

5.4.1 Linéarité de quelques difféomorphisms

Pour étudier la linéarité de certains difféomorphisms on peut se baser sur le théorème suivant :

Théorème 5.4.1 [164] : *Soit $\phi : E_1 \longrightarrow E_2$ un isogénie.*
Alors : $\phi(P+Q) = \phi(P) + \phi(Q)$ pour tout P,Q $\in E_1$.

Prouver la linéarité revient à prouver tout d'abord l'isogénie.
Nous proposons, ci-après, une méthode par récurrence pour prouver la linéarité.

Premier cas : Linéarité de quelques distorsions

En utilisant la méthode par récurrence, dire qu'un difféomorphism ϕ est linéaire revient à montrer que $\phi(aP)$ = a$\phi(P)$ pour tout a > 0.

Les distorsions $\phi(x, y) = (-x, iy)$ et $\phi(x, y) = (\zeta x, y)$ sont linéaires.

Essayons de montrer la linéarité pour $\phi(x, y) = (-x, iy)$.

D'abord, la propriété est vraie pour a = 1. On doit montrer qu'il est linéaire pour tout a > 0.

Supposons qu'elle est vrai pour a-1 càd : $\phi((a-1)P)$ = $(a-1)\phi(P)$ et on cherche à monter que $\phi(aP)$ = $a\phi(P)$

Démontrer cela revient à prouver que :

$$\phi(P + (a-1)P) = \phi(P) + \phi((a-1)P)$$

On a :

$$P + (a-1)P = (x_P, y_P) + (x_{(a-1)P}, y_{(a-1)P}) = aP$$

Avec :

$$\lambda_{aP} = \frac{y_{(a-1)P} - y_P}{x_{(a-1)P} - x_P}, x_{aP} = (\lambda_{aP})^2 - x_P - x_{(a-1)P}$$

et

$$y_{aP} = \lambda_{aP}(x_P - x_{aP}) - y_P$$

Dans l'autre partie, on a :

$$\phi(P) + \phi((a-1)P) = (-x_P, iy_P) + (-x_{(a-1)P}, iy_{(a-1)P})$$

Avec :

$$\lambda_{\phi(P)+\phi((a-1)P)} = \frac{iy_{(a-1)P} - iy_P}{-x_{(a-1)P} + x_P} = -i\lambda_{aP}$$

Alors :

$$x_{\phi(P)+\phi((a-1)P)} = (-i\lambda_{aP})^2 + x_P + x_{(a-1)P} = -x_{aP}$$

et

$$y_{\phi(P)+\phi((a-1)P)} = -i\lambda_{aP}(-x_P + x_{aP}) - iy_P = iy_{aP}$$

Par conséquent :

$$\phi(P) + \phi((a-1)P) = (-x_{aP}, iy_{aP})$$

Et puisque :

$$\phi(P + (a-1)P) = \phi(aP) = (-x_{aP}, iy_{aP})$$

On a bien :

$$\phi(P + (a-1)P) = \phi(P) + \phi((a-1)P)$$

On a supposé que :

$$\phi((a-1)P) = (a-1)\phi(P)$$

Alors :

$$\phi(aP) = \phi(P + (a-1)P) = \phi(P) + \phi((a-1)P) = \phi(P) + (a-1)\phi(P) = a\phi(P).$$

Cela montre clairement la linéarité de cette distorsion.
En utilisant la méme méthode, on peut démontrer la linéarité des autres distorsions (par exemple celle de : $\phi(x, y) = (\zeta x, y)$).

Second cas : Linéarité de φ_d

Il est simple de montrer la linéarités de tous les isomorphisms déclarés dans le tableau 5.2. Par exemple, montrons la propriété pour φ_2.
De même que précédemment, nous utilisons la méthode de récurrence.
Ici, nous donnons que les grandes lignes et il sera simple d'extraire le reste. Il suffit alors de démontrer que :

$$\varphi_2(P + Q) = \varphi_2(P) + \varphi_2(Q)$$

Pour la faire, on a :

$$\lambda_t = \frac{v^{-(\frac{3}{4})}}{v^{-(\frac{1}{2})}}\lambda = v^{-(\frac{1}{4})}\lambda$$

Donc, on a bien :

$$x_{(P+Q)_t} = (v^{-(\frac{1}{4})}\lambda)^2 - v^{-(\frac{1}{2})}x_P - v^{-(\frac{1}{2})}x_Q = v^{-(\frac{1}{2})}x_{P+Q}$$

Et

$$y_{(P+Q)_t} = v^{-(\frac{1}{4})}\lambda(v^{-(\frac{1}{2})}x_P - v^{-(\frac{1}{2})}x_{P+Q}) - v^{-(\frac{3}{4})}y_P = v^{-(\frac{3}{4})}y_{P+Q}$$

Avec le fait que les paramètres : λ_t, $x_{(P+Q)_t}$, x_{P_t}, x_{Q_t} caractérisent les points de la tordue.

5.4.2 Simple attaque de CDHP

Le but devant cette partie est de préciser les conditions à tenir en compte pour rendre CDHP plus sécurisé. En utilisant l'outil Tate, Frey et Rück [78] ont projeté le problème du logarithme discret sur les courbes elliptiques à celui du corps fini. Nous utilisons le même outil qui est le couplage de Tate afin de faire une autre sorte de cryptanalyse [9], sous laquelle nous fixerons certaines conditions qui permettent de rendre CDHP plus sécurisé.

Étant donné P, aP et bP, essayons de calculer abP. Choisissons le couplage de Tate et procédons comme suit :
Après avoir calculé l'ordre des points P, aP, bP et en utilisant l'expression du couplage, on a :

$$t_{r_{aP}}(aP, bP) = (f_{D_{aP}}(D_{bP}))^{(q^k-1)/r_{ab}} \tag{5.38}$$

Mais, cette expression est peut être triviale, selon la propriété suivante :

Propriété 5.4.1 (Galbraith [132])
Soit $P \in E(F_p)[r]$, avec r relativement premier avec p. Alors :

$$t_r(P, P) \neq 1 \quad si \quad k = 1 \quad et \quad t_r(P, P) = 1, \quad pour \quad k \ > 1. \tag{5.39}$$

Par conséquent : $t_{r_{aP}}(aP, bP) = t_r(P, P)^{ab} = 1$, quelque soit les entiers a et b, ce qui rend ce calcul trivial. Ce problème peut être résolu d'après la propriété suivante :

Propriété 5.4.2 (Veurheul[132])
Soit r un entier premier, $P \in E(F_p)[r]$ et $Q \in E(F_{p^k})$ deux points de la courbe elliptique qui sont linéairement indépendants et k > 0. Alors $t_r(P, P)$ est non-dégénéré càd $\neq$ de 1

Pour projeter ce résultat sur notre cas, nous choisissons l'une des difféomorphisms annoncés précédemment (le plus convenable à la courbe elliptique choisie). Alors, on a :

$$t_{r_{aP}}(aP, \phi(bP)) = t_{r_P}(P, a\phi(bP)) = t_{r_P}(P, \phi(abP)) \tag{5.40}$$

(Par la bilinéarité des couplages et la linéarité des difféomorphisms). Cette égalité est non triviale, puisque aP, $\phi(bP)$ et P, $\phi(abP)$ sont linéairement indépendants.

Alors : $t_{r'}(aP, \phi(bP))$ = $t_r(P, \phi(abP))$ est non trivial (r' et r sont respectivement l'ordre de aP et P). Ce qui implique que :

$$(f_{D_{aP}}(D_{\phi(bP)}))^{\frac{q^k-1}{r'}} = (f_{D_P}(D_{\phi(abP)}))^{\frac{q^k-1}{r}} \tag{5.41}$$

Alors :

$$(f_{D_{aP}}(D_{\phi(bP)}))^{\frac{r}{r'}} = f_{D_P}(D_{\phi(abP)}) \tag{5.42}$$

Exploitation de l'idée

Comme on a :

$$f_r = \frac{(P)(P)}{(2P)}\frac{(P)(2P)}{(3P)}...\frac{(P)((r-1)P)}{(rP)} = f_{D_P} = f_P \tag{5.43}$$

Nous proposons de simplifier cette expression. Dans le numérateur figurent les lignes qui lient deux points différents, par exemple, $L_{(P),(2P)} = (P)(2P)$ est une ligne qui lie deux points P et 2P. Tandis qu'au dénominateur on trouve la ligne vertical qui passe par le point de la tangente ($V_{2P} = (2P)$ est la ligne verticale qui passe par 2P). Avec Maple ou un autre logiciel, on calcule P, 2P ,....., rP. Après il serait, ensuite, simple de trouver l'équation correspondante de chaque numérateur, et celle de chaque dénominateur. En utilisant un programme (ou la main si r est petit), on peut simplifier l'expression de f_P tenant en compte que $Y^2 = X^3 + eX + f$ (équations de la courbe elliptique choisie). On simplifie chaque degré de Y qui dépasse 2, à la fin on obtient une formule sous la forme :

$$f_P = \frac{f_{P_{nu_X}} + Y f_{P_{nu_Y}}}{f_{P_{de_X}} + Y f_{P_{de_Y}}} \tag{5.44}$$

Avec $f_{P_{nu_X}}$ et $f_{P_{nu_Y}}$ sont dans K[X] (K est le corps choisi) elles ont des degrés inférieurs ou égaux à r.

Retournons à (5.42), on associe à $\phi(abP)$ les coordonnées $(X_{\phi(abP)}, Y_{\phi(abP)})$. Par Miller, on calcule $f_{D_{aP}}(D_{bP})$ après avoir remplacé D_{bP} par son expression, par exemple, $D_{bP} = [bP + R] - [R]$ où R est selon notre choix.

Il est simple d'appliquer l'algorithme de Miller à $f_{D_{aP}}(D_{bP})$, puisque aP, bP sont donnés. Puis, on calcule : $f_{D_{aP}}(D_{bP})^{\frac{r}{r'}}$.

Mais, $D_{\phi(abP)} = [\phi(abP)] - [O]$, remplaçons cette expression dans (5.42), on trouve alors :

$$f_{D_P}(\phi(abP)) = (f_{D_{aP}}(D_{bP})^{\frac{r}{r'}})(f_{D_P}(O)) \tag{5.45}$$

Aussi, selon l'algorithme de Miller, on calcule $f_{D_P}(O)$. Par conséquent, on a :

$$f_{D_P}(\phi(abP)) = cte_1 \tag{5.46}$$

Remplaçons $\phi(abP)$ par ses coordonnées dans l'expression de f_{D_P} qui est (5.44) pour trouver : $Y_{\phi(abP)}$ en fonction de $X_{\phi(abP)}$ en utilisant (5.46). Ensuite, substituons cette expression dans $Y^2 = X^3 + eX + f$, ce qui donne une équation de degré inférieur ou égal à r, sa solution est $X_{\phi(abP)}$ qu'on cherche. En succédant cette solution dans l'équation de la courbe elliptique, conduit à trouver l'expression de $Y_{\phi(abP)}$.
Ces coordonnées sont exactement les coordonnées de $\phi(abP)$ qu'on cherche. Après, il sera simple d'extraire abP.
Mais, pour des r plus grands que 12, on ne peut pas résoudre une équation de tel degré. On propose de suivre la méthode suivante :
Nous discutons cette méthode selon les cas :
1^{er} cas : Supposons que r est paire càd r=$2r''$ et retournons à :

$$t_{r'}(aP, \phi(bP)) = t_r(P, \phi(abP)) \tag{5.47}$$

Exposons par r'', on trouve alors :

$$(t_{r'}(aP, \phi(bP)))^{r''} = (t_r(P, \phi(abP)))^{r''} \tag{5.48}$$

Cette formule est équivalente à :

$$(t_{r'}(aP, \phi(bP)))^{r''} = (t_r(r''P, \phi(abP))) \tag{5.49}$$

Alors :

$$(f_{D_{aP}}(D_{bP}))^{r''} = f_{D_{r''P}}(D_{\phi(abP)}) \tag{5.50}$$

En procédant comme précédemment, on trouvera que :

$$f_{D_{r''P}}(\phi(abP)) = (f_{D_{aP}}(D_{bP}))^{r''}(f_{D_P}(O)) = cte_2 \tag{5.51}$$

Mais, puisque $f_{D_{r''P}}$= $\frac{(r''P)(r''P)}{(2r''P)}$ et $V_{2r''P} = 1$, d'après le fait que : 2r"P=O. Alors :

$$f_{D_{r''P}} = f_{r''P} = y - m_{r''P}x - \beta_{r''P} \tag{5.52}$$

Avec le fait que :

$$\beta_{r''P} = y_{r''P} - m_{r''P} \cdot x_{r''P} \ \ et \ \ m_{r''P} = \frac{3x_{r''P}^2 + 1}{2y_{r''P}} \tag{5.53}$$

Remplaçons dans l'expression (5.52) le point $\phi(abP)$ par ses coordonnées $y_{\phi(abP)}$ et $x_{\phi(abP)}$. À la fin, on substitue l'expression trouvée dans (5.51) pour exprimer par exemple $y_{\phi(abP)}$ en fonction de $x_{\phi(abP)}$ (ou l'inverse). Après une substitution dans l'expression développée, on trouve simplement une équation (lié à $x_{\phi(abP)}$) de degré trois qui peut être résolue facilement, soit par la main ou par un logiciel. Une fois $x_{\phi(abP)}$ est calculé, on le remplaçe dans l'équation de la courbe elliptique pour extraire $y_{\phi(abP)}$.

Alors les coordonnées ($x_{\phi(abP)}$,$y_{\phi(abP)}$) sont exactement les coordonnées de $\phi(abP)$, après il sera simple d'extraire abP.

$\underline{2^{ème} \text{ cas}}$: r est impair, c'est le cas le plus fréquent ; pour le résoudre, nous suivons ce qui suit :

Nous le discutons aussi selon les cas.

Premier cas : r est composé càd il est non premier, alors on a : r = $r''r'''$. Effectivement, l'un de ces deux facteurs est grand, supposons que c'est r'' (alors r''' est petit)

Donc, exposons par r'', on a :

$$(t_{r'}(aP, \phi(bP)))^{r''} = (t_r(P, \phi(abP)))^{r''} \tag{5.54}$$

Ce qui implique que :

$$f_{r'''}(D_{\phi(abP)}) = (t_{r'}(aP, \phi(bP)))^{r''} \tag{5.55}$$

Alors, si on arrivé à simplifier

$$f_{r'''} = \frac{(r''P)(r''P)}{(2r''P)} \frac{(r''P)(2r''P)}{(3r''P)} ..... \frac{(r''P)((r''r''' - 1)P)}{(r''r'''P)} \tag{5.56}$$

On peut obtenir le résultat souhaité ; mais il est lié à r'''

Sinon, on peut faire autrement (en poursuivant l'autre cas) :

Second cas : Si r est non composé càd r est premier, il est le cas le plus délicat et le plus difficile d'être casser.

On cherche un point Q dans E(k), qui n'est pas dans E [r], sous la condition que P-Q et Q ont des ordres pairs. Si on a cela, on peut résoudre bien ce problème, puisque :

Supposons par exemple que l'ordre de P-Q est 2s et celui de Q est $2s'$. En plus, supposons par exemple que $s > s'$ alors $s = s's''$ (voir plus tard le rang de E).

Retournons maintenant à l'équation :

$$t_{r'}(aP, \phi(bP)) = t_r(P, \phi(abP)) \tag{5.57}$$

Alors :

$$t_{r'}(aP, \phi(bP)) = t_r(P - Q + Q, \phi(abP)) = t_{2s}(P - Q, \phi(abP))t_{2s'}(Q, \phi(abP)) \tag{5.58}$$

En exposant par s ce qui implique que :

$$(t_{r'}(aP, \phi(bP)))^s = (t_{2s}(P - Q, \phi(abP)))^s(t_{2s'}(Q, \phi(abP)))^s \tag{5.59}$$

Le terme :

$$(t_{2s}(P - Q, \phi(abP)))^s = t_{2s}(s(P - Q), \phi(abP)) = f_{s(P-Q)}(D_{\phi(abP)}) \tag{5.60}$$

est simplifiable (c'est une équation de deux variables et de degré 1).
Pour le second terme càd $(t_{2s'}(Q, \phi(abP)))^s$, on a :
Si s'' est divisible par 2, alors on a : $(t_{2s'}(Q, \phi(abP)))^s = 1$, sinon, cela dépend de s'', on peut trouver l'expression désirée (plus simplifiée), sinon on change le Q.
Enfin, puisque : $(t_{r'}(aP, \phi(bP)))^s$ = cte et que aP et bP sont connus. Alors, en poursuivant la méthode précédente on peut extraire les coordonnées $(x_{\phi(abP)}, y_{\phi(abP)})$.

Existence de P-Q et Q munis des ordres désirés

Soit E une courbe elliptique définie sur F_p, donc E(F_p) est un groupe commutatif de rang égal à 1 ou 2 [*103*]. Alors : E(F_p) $\simeq Z_{n_1} \oplus Z_{n_2}$ avec $n_1 \setminus n_2$ et $n_1 \setminus$ p-1.

L'algorithme suivant sert au calcul de n_1 et n_2.

Deuxième algorithme de Miller :

1. Calculer n = card(E(K)) (en utilisant l'algorithme de Schoof ou l'un de ses variantes).
2. Prendre arbitrairement u, v dans E.
3. Calculer s = ord(u) ; t = ord(v) (pour faire cela on doit connaître la factorisation de n).
4. Calculer m = ppcm(s,t) et ζ = $e_m(u, v)$ une racine de l'unité
5. Calculer d = ord(ζ), vérifier si md = n.
6. Si c'est vrai alors n_1= d, n_2 = m. Sinon retourner à 2.

Alors on peut choisir P-Q et Q des points d'ordres n_1 et n_2.

Il y a une forte probabilité qu'on peut trouver n_1 et n_2 paires. En effet, le n_1 est peut-être paire (p-1 est paire), il suffit alors de chercher uniquement les u et v correspondants qui donnent cette condition et si on les trouve, automatiquement le n_2 sera aussi paire.

Concrétisation de l'idée

Pour mieux comprendre notre idée, nous proposons l'exemple ci-dessous.

Choisissons la courbe $y^2 = x^3 + 1$ sur le corps F_{11}, P=(2,3) est bien un point de cette courbe. Pour appliquer la méthode déclarée juste au dessus, on doit trouver l'ordre de P.

Comme 2P=P+P=(2,3)+(2,3). On a : $m_{P,P}$ = $\frac{3x^2}{2y} = \frac{3\times4}{6} = \frac{12}{6} = 2$. Donc : $x_{2P} = 2^2 - 4 = 0$ et $y_{2P} = 2(2) - 3 = 1$ alors : 2P = (0,1)

3P=2P+P=(0,1)+(2,3), $m_{2P,P}$ = $\frac{3-1}{2-0}$=1 ; $x_{3P} = 1^2 - 0 - 2 = -1 = 10$ et $y_{3P} = 1(2-10) - 3 = -11 = 0$. Alors 3P = (10,0)

4P=2P+2P=(0,1)+(0,1), $m_{2P,2P}$ = 0. Donc : $x_{4P} = 0$ et $y_{2P} = 10$. Alors : 4P = (0,10)

5P=4P+P=(0,10)+(2,3), $m_{4P,P}$ = $\frac{10-3}{0-2}$=$\frac{7}{9} = \frac{7\times5}{9\times5} = 2$; $x_{5P} = 2^2 - 0 - 2 = 2$ et $y_{5P} = 2(0) - 3 = 8$. Alors : 5P = (2,8)

Pour 6P = 5P+P, on a : $x_{5P} = x_P = 2$ et $y_{5P} \neq y_P$. Nécessairement 6P = O.

Par conséquent, P à un ordre 6. On a, 6 ne divise pas 11 ; mais $6 \setminus 11^2 - 1$. Alors on peut regrouper les calculs dans le corps F_{11^2}.

Notre but devant cet exercice est de trouver le point 20P à partir des deux points 4P et 5P. En utilisant le difféomorphism $\phi(x, y) = (\zeta x, y)$ qui est convenable à la courbe utilisée, alors on a (selon les propriétés : bilinéarité du couplage et la linéarité de distorsion) :

$$t_3(4P, \phi(5P)) = t_6(P, 4\phi(5P)) = t_6(P, \phi(20P))$$

Comme P à un ordre pair 6=2.3, alors en exposant par 3 on trouve :

$$t_3(4P, \phi(5P))^3 = t_6(P, \phi(20P))^3$$

Puisque :

$$\zeta = \frac{11-1}{2}(1 + 3^{(\frac{11+1}{4})}) = 5(1 + 5i) = 5 + 3i$$

Alors :

$$\phi(5P) = (\zeta x_{5P}, y_{5P}) = ((5 + 3i)2, 8) = (10 + 6i, 8)$$

Sans passer par l'algorithme de Miller, on peut calculer : $t_6(4P, \phi(5P))^3$, puisque : $t_6(12P, \phi(5P))$ = $t_6(O, \phi(5P)) = 1$, comme f_0 = 1. Donc : $f_{3P}(D_{\phi(20P)}) = 1$

Mais, 3P à l'ordre 2. Alors : $f_{3P} = f_2 = \frac{(3P)(3P)}{6P}$ = $\frac{T_{3P}}{V_{6P}}$

Comme : 6P = O, $V_{6P} = 1$ et $T_{3P} = V_{3P} = x - x_{3P}$ = x-10 = x+1. En plus on a :

$$f_{3P}(D_{\phi(20P)}) = \frac{f_{3P}(\phi(20P)}{f_{3P}(O)} = 1$$

Comme : $f_{3P}(O)$ = 1 alors $f_{3P}(\phi(20P)) = 1$. Donc : $x_{\phi(20P)} + 1 = 1$ implique $x_{\phi(20P)} = 0$. Ce qui implique que : $y_{\phi(20P} = 1$ ou $y_{\phi(20P} = -1 = 10$ puisque $y^2 = x^3 + 1$

Par conséquent, $\phi(20P) = (0, 1)$ ou $\phi(20P) = (0, 10)$. Ainsi, $20P = (0, 1)$ ou $20P = (0, 10)$ puisque : $\phi(20P) = (\zeta\ x_{20P}, y_{20P})$ donc : $20P = (\zeta^2\zeta\ x_{20P},\ y_{20P})$ = (x_{20P} ,y_{20P}) or $\zeta^3 = 1$. On a alors : $20P = (0, 1)$ ou 20P = (0,10) =4P.

Le cas (0.10) est exclu, puisqu'on ne peut pas trouver à partir de aP et bP un abP = aP ou à bP.

Alors : 20P = (0,1) = 2P ce qui est vrai, comme 20P = 18P+ 2P= O+2P = 2P

5.4.3 De BDHP à CDHP

Toujours avec le couplage de Tate, on peut répondre [9] à la question posée dans l'article de Boneh et Franklin [22] et qui reste sans réponse absolue jusqu'à nos jours : peut-on aller de BDHP à CDHP ?

Oui, la réponse se résume ainsi :

Étant donné P, aP, bP, cP et $t(aP, bP)^c = cte$ essayons de trouver par exemple abP

Comme :

$$t(aP, bP)^c = t(P, P)^{abc} = t(P, P)^{cab} = t(cP, abP) = cte \tag{5.61}$$

Si cP à un ordre petit r (cP peut être grand alors son ordre est peut être petit), donc on a :

$$t(cP, abP) = f_{D_{cP}}(D_{abP}) = cte \tag{5.62}$$

Dans ce cas, nous calculons par Miller uniquement $f_{D_{cP}}(O)$ alors :

$$f_{D_{cP}}(abP) = cte f_{D_{cP}}(O) = CTE \tag{5.63}$$

On peut écrire $f_{D_{cP}}$ sous la forme :

$$f_r = \frac{(cP)(cP)}{(2cP)}\frac{(cP)(2cP)}{(3cP)}.....\frac{(cP)((r-1)cP)}{(rcP)} = f_{D_{cP}} = f_{cP} \tag{5.64}$$

Avec la même procédure mentionnée dans 5.4.2, on simplifie l'expression $f_{D_{cP}}$ afin d'obtenir une expression sous la forme :

$$f_{cP} = \frac{f_{cP_{nu_X}} + Y f_{cP_{nu_Y}}}{f_{cP_{de_X}} + Y f_{cP_{de_Y}}} \tag{5.65}$$

On remplaçe les coordonnées de abP=(X_{abP}, Y_{abP}) dans cette expression
D'après (5.63) on peut obtenir $Y_{\phi(abP)}$ en fonction de $X_{\phi(abP)}$. Dans l'équation de la courbe on remplaçe le Y_{abP} et on trouve alors une équation de degré au plus r (sa solution dépend de la valeur de r, on peut accepter une équation de degré < 20). La solution de cette équation est bien $X_{\phi(abP)}$ qu'on cherche, après on calcule $Y_{\phi(abP)}$.
Mais si r est non petit, selon la parité de r et suivant la méthode annoncée dans la partie 5.4.2, on peut extraire ($X_{\phi(abP)}$,$Y_{\phi(abP)}$) (le (X_{abP},Y_{abP}) sera ainsi simple).

5.4.4 Remarques importantes

La première remarque qu'on peut signaler est qu'avec le couplage de Weil, on ne peut attaquer ni CDHP ni BDHP, puisque :
Si :

$$e_r(aP, \phi(bP)) = e_r(P, \phi(abP)) \tag{5.66}$$

Alors :

$$\frac{f_{D_{\phi(bP)}}(D_{aP})}{f_{D_{aP}}(D_{\phi(bP)})} = \frac{f_{D_{\phi(abP)}}(D_P))}{f_{D_P}(D_{\phi(abP)})} \tag{5.67}$$

Donc :

$$f_{D_{\phi(bP)}}(D_{aP}) \times f_{D_P}(D_{\phi(abP)}) = f_{D_{\phi(abP)}}(D_P) \times f_{D_{aP}}(D_{\phi(bP)}) \tag{5.68}$$

Il est bien clair que dans l'équation (5.68), figure deux inconnus f_{abP} et D_{abP}. Ce qui est irrésoluble.
Par contre, avec le couplage de Tate (et ses variantes) on peut bien arriver à ce but ; mais cela dépend des conditions suivantes :

1. Dans le cas des couplages symétriques, il faut utiliser des courbes qui fonctionnent avec des isomorphisms :
   - Courbes supersingulières puisqu'elles fonctionnent avec les distorsions.
   - Courbes ordinaires flexibles aux tordues et cela dépend de leur degré de plongement.
2. La parité de l'ordre du point utilisé comme premier argument dans le calcul du couplage, un point d'ordre premier est privilège pour résister à l'attaque de la partie 5.4.2.
3. L'ordre de la courbe tout entier, une courbe d'ordre composé peut bien servir la cryptanalyse (deuxième algorithme de Miller).

Ces trois conditions [9] constituent ainsi les piliers de la sécurité de CDHP et BDHP (ainsi des IBE), qu'il faut tenir en compte.

Une autre remarque qu'on peut ajouter dans cette partie concerne une comparaison entre MOV et FR, laquelle de ces deux attaques servira la cryptanalyse ?

Attaque MOV [134] ou celle de FR [78] ?

Menezes, Okamoto et Vanstone [134] utilisent le couplage de Weil pour projeter le logarithme discret des courbes elliptiques sur les corps finis, un an après, en 1994, Frey et Rück [78] ont utilisé le couplage de Tate pour faire la même procédure.

Algorithme 5.4.1 : réduction de MOV	***Algorithme 5.4.2 : réduction de FR***
Entrée : P $\in$ E(F_p) d'ordre r et Q $\in \prec P \succ$	**Entrée :** P $\in$ E(F_p) d'ordre r
Trouver k minimal tel que : E[r] $\in E(F_{p^k})$	Trouver k minimal tel que : $\mu(r) \in E(F_{p^k})$
Trouver R $\in$ E[r] et calculer : $\alpha = e_r(P, R)$	Trouver R $\in E(F_{p^k})$ tel que : R n'appartient pas à rE(F_{p^k} ($t_r(P, Q)$ est non trivial)
Calculer $\beta = e_r(Q, R)$	Calculer $\alpha = t_r(P, R)^{(p^k-1)/r}$ et $\beta = t_r(Q, R)^{(p^k-1)/r}$
Calculer le logarithme discret de β versus α dans F_{p^k}	Calculer l le logarithme discret de β versus α dans F_{p^k}
Retourner l'entier l tel que Q=lP	**Retourner** l'entier l tel que : Q=lP

TABLE 5.4 – Principe des attaques MOV et FR

Pour les deux couplages Weil et Tate, le EDLP (extraire r à partir de rP) peut être projeté à PDL (extraire r à partir de g^r) puisque : e(rP,Q) = $e(P, Q)^r$ =g^r où g=e(P,Q).
Nous constatons que la deuxième méthode où on utilise Tate, aide aux bons résultats que la première. En effet, le couplage de Tate est deux fois plus rapide que celui de Weil [13], en plus, dans l'expression du couplage de Weil il figure deux fois celle de Tate. Alors, dans des niveaux de sécurité équivalentes, extraire l de l'algorithme FR sera facile que de l'extraire de celui de MOV.

5.5 Courbes adaptées aux calculs des couplages

5.5.1 Méthode de Multiplication Complexe

Pour construire des courbes qui vérifient certaines propriétés, on passe en général par la méthode de multiplication complexe. Cette méthode a été premièrement utilisée pour construire les courbes elliptiques sur C, elle a été adaptée en 1993 par Atkin et Morain [11] dans le cas des corps finis. Le principe de la méthode se résume ainsi :

- Générer les paramètres p, t, après avoir résolu une équation sous la forme : $b^2 - 4a$ par la méthode de Pell généralisée, en réalité, l'équation cherchée à résoudre est : $4p - t^2 = Dy^2$ où -D est le discriminant fondamental, ceci d'après le principe suivant : Étant donné p un nombre premier et un entier t dans la borne de Hasse, trouver une courbe E définie sur F_p qui a p + 1 - t points. L'idée consiste de retrouver les solutions de : $X^2 - tX + p = 0$, d'après le polynôme caractèristique de l'endomorphism de Frobenius sur $E(F_p)$. Si on considère D le discriminant de cette équation, la méthode naïve dit que : trouver une solution à X^2 - tX + p = 0 revient à résoudre 4p - $t^2 = Dy^2$.
  L' équation de Pell génère les paramètres $(x_p + \sqrt{3D}y_p)(x_0 + \sqrt{3D}y_0)^k$ avec k $\in$ Z. Chaque fois nous donnons un k, nous tombons sur des nouveaux p (à condition que p soit premier) et t (voir annexe C).

- Après avoir trouvé le p désiré et fixer le D tel que : D $\neq$ -1, ce D satisfait deux cas :
  — D = 3mod 4 et sans facteurs carrés, ou
  — Soit D = 4m avec m = 1 ou = 2mod 4 sans aussi facteurs carrés.
  Ensuite, essayer de calculer le polynôme de Hilbert $H_{-D}(X)$ (voir [35][130] pour une culture sur ce polynôme).

- Une fois trouver $H_{-D}(X)$ et p, essayer de trouver les racines de ce polynôme dans F_p, nous le notons par exemple par j_1 (si on a 1, ça peut exister deux, trois...), calculer ensuite $k_1 = \frac{j_1}{1728 - j_1}$ pour un $j_1 \neq 0$ et 1728. La courbe cherchée à une équation sous la forme : $y^2 = x^3 + 3k_1c^2x + 2k_1c^3$ où c est quelconque dans F_p.

Équation de Pell

Pour résoudre cette équation, il faut chercher une solution minimale (x_0, y_0) qui vérifie : ${x_0}^2 - 3D{y_0}^2$ = 1. Pour chercher la solution minimale, nous devrons faire des calculs suivant les réductions du développement en fractions continues de $\sqrt{3D}$.
L'algorithme 2 (Listing 2) donné dans l'annexe résume la staratégie des fractions continues pour résoudre $x^2 - 3Dy^2$ = 1.

Équation de Pell généralisée

Pour résoudre une équation de Pell généralisée : $x^2 - 3Dy^2 = m$, dans le cas où 3D > m (pour l'inverse voir [130]), nous cherchons tout d'abord la solution de $x^2 - 3Dy^2 = 1$ suivant l'algorithme cité dans l'annexe (listing 2), puis, nous cherchons une solution particulière de $x^2 - 3Dy^2 = m$ par brute force.

À cause des deux attaques MOV [134] et FR [78], les courbes supersingulières sont non désirées (degrés de plongement très petits) dans la sécurité de la cryptographie basée sur les couplages. C'est pourquoi, il est recommandé de construire des courbes adaptées au calcul d'un couplage, qui sont connues en général par courbes **ordinaires**.

5.5.2 Construction des courbes adaptées aux calculs des couplages

En tenant en compte, l'attaque de Pohling Hellman, la forme des couplages ainsi que la paramétrisation des courbes elliptiques, une courbe adaptée au calcul du couplage doit satisfaire le système suivant :

$$\begin{cases} r|p+t-1 & \\ r|p^k-1, & \text{avec r et p sont premiers} \\ Dy^2=4p-t^2, & \text{pour un entier donné y} \end{cases}$$

Le paramètre k s'appelle le degré de plongement (où k $\leq$ 50 comme c'est précisé dans [79]), nous disons que la courbe construite E est bien adaptée si elle vérifie les deux propriétés suivantes :

1. Il existe un nombre premier r $\geq \sqrt{p}$ qui divise $\neq E(F_p)$
2. Le degré de plongement de E suivant r est inférieur à $log_2(r)/8$.

Les courbes adaptées au calcul du couplage sont à peu près nombreuses, nous marquons MNT [139] et Cocks Pinch [59] comme premières réponses aux attentes dessus.
Nous rappelons, ci-après, ces deux courbes, nous ajoutons les autres mentionnées dans la littérature et qui répondent aussi aux expectatives dessus.

Courbes de MNT :

En 2001 Miyaji, Nakabayashi et Takano ont exploité la courbe suivante :

Théorème 5.5.1[139] : *Soit p un nombre premier, et soit E/F_p une courbe ordinaire telle que r =* $| E(F_p) |$ *est premier. Soit t=p+1-r.*

1. *E à un degré de plongement k=4 si et seulement s'il existe x $\in$ Z tel que t=-x ou t=x+1* $p = x^2 + x + 1$
2. *E à un degré de plongement k=3 si et seulement s'il existe x $\in$ Z tel que t=-1$\pm$6x et* $p = 12x^2 - 1$
3. *E à un degré de plongement k=6 si et seulement s'il existe x $\in$ Z tel que t=1$\pm$2x et* $p = 4x^2+1$

Après avoir obtenu les paramètres p, t, nous suivons la méthode de multiplication complexe pour construire la courbe. L'avantage de cette méthode est qu'elle permet de générer des courbes d'ordres premiers.
Lors d'un manuscrit non publié et pendant la même année 2001, Cocks et Pinch donnent une autre proposition.

Méthode de Cocks-Pinch :

Théorème 5.5.2 [59] : *Fixer un entier positive k et un D positive et sans facteurs carrés. Exécuter les étapes suivantes :*

1. *Soit r un nombre premier tel que : k|r-1 et -D|r*
2. *Soit z un k-ème racine de l'unité dans* $(Z/rZ)^{\times}$ *(ce z existe puisque k|r-1). Soit t'=z+1*
3. *Soit y'=(t'-2)/*$\sqrt{-D}$*(mod r)*
4. *Soit t'* $\in$ *Z congrue à t mod r, et soit y* $\in$ *Z congrue à y' mod r. Soit p=*$(t^2 + Dy^2)/4$

De même nous suivons la méthode de multiplication complexe pour construire la courbe, après avoir obtenu p et t.

Courbes de Brezing et Weng

Théorème 5.5.3 [43] : *Fixer un entier positif k et un nombre sans facteur D. Exécuter les étapes suivantes :*

1. *Trouver un polynôme irréductible r(x)* $\in$ *Z[x] avec des coefficients positifs tel que : K = Q[x]/(r(x)) est un corps contenant* $\sqrt{-D}$ *et un corps cyclotomique* $Q()$*.*
2. *Choisir une k-ème racine de l'unité* $\zeta_k \in K$
3. *Soit t (x)* $\in$ *Q[x] un polynôme qui projette dans* ζ_k *+1 dans K.*
4. *Soit y(x)* $\in$ *Q[x] qui project dans (*ζ_k *- 1) /* $\sqrt{-D}$ *dans K.*
5. *Soit p(x)* $\in$ *Q[x] donné par* $(t(x)^2 + Dy(x)^2)/4$*.*

L'avantage que représente cette méthode est que les paramètres sont représentés comme des polynômes. Il suffit de donner des valeurs à x et de tester ensuite la parité de ces paramètres. Alors, on déroule une seule fois l'algorithme et on change uniquement les valeurs de x afin d'obtenir des valeurs attendues de p et t. Par contre, avec Cocks-Pinch, il faut chaque fois dérouler l'algorithme.
La méthode qui permet de tester la parité d'un polynôme est comme suit :

Définition 5.5.1 : *Soit P(x) un polynôme qui admet des coefficients rationnels. On dit que P représente un premier, si elle vérifie les hypothèses suivantes :*

1. *P(x) est non constante.*
2. *P(x) à des coefficients positifs.*
3. *P(x) est irréducible.*
4. *P(x) ∈ Z quelques soit x ∈ Z (ça sera mieux pour un nombre infini de Z)*
5. *gcd(P(x) : x, P(x) ∈ Z) = 1.*

Toujours avec la méthode de la multiplication complexe (résoudre cette fois $4p(x)-t(x)^2 = Dy^2$), nous cherchons la courbe convenable et qui a comme paramètres p et t (après un remplacement par des valeurs x qui conduisent aux p premiers) obtenus par Brenzing-Weng.

Méthode de Freeman [80]

En utilisant une factorisation faite par Galbraith et al [84], Freeman (voir [80] pour plus de détails) a trouvé la paramétrisation suivante :

- t(x) = $10x^2$ + 5x +3
- r(x) = $25x^4 + 25x^3 + 15x^2 + 5x + 1$
- p(x) = $25x^4 + 25x^3 + 25x^2 + 10x + 3$

Aussi, le remplacement par des x convenables, nous permet d'obtenir des p et r premiers. L'avantage de cette courbe est que son degré de plongement est 10 (grand paramètre), en plus, il permet de générer des courbes d'ordre (cardinal) premier.

Courbes de Barreto-Naehrig [17]

Pour un degré de plongement k=12 et en utilisant les paramètres ci-dessous, Barreto-Naehrig ont généré une courbe d'ordre premier.

- t (x) = $6x^2 + 1$
- $r(x) = 36x^4 + 36x^3 + 18x^2 + 6x + 1$
- $p(x) = 36x^4 + 36x^3 + 24x^2 + 6x + 1$

Un algorithme qui permet de générer d'une façon simple les paramètres de la courbe est cité dans [17] (voir aussi l'annexe, Listing 1).

Autres courbes adaptées aux couplages sont citées dans l'état de l'art de Freeman et al [79]. Nous nous limitons à ces cinq courbes pour raison que MNT, BN et Freeman permettent de construire des courbes d'ordres premiers. Tandis qu'avec Cocks-Pinch et Brezing-Weng, on peut fixer le degré de plongement k au début. Nous verrons l'utilité de ces deux conditions dans la partie 5.6.3.

5.6 Implémentation des IBE

Tout ce qu'on expose jusqu'ici rentre dans le cadre théorique. Nous arrivons maintenant à la partie pratique, son organisation sera comme suit :
Dans la partie 5.6.2 nous traitons l'implémentation du cryptosystème sous le report 367/2012 (quatrième schéma pour l'IBE sous le modèle Random Oracle), cela a pour but de mettre en évidence ce nouveau schéma. Dans la partie 5.6.3 nous effectuions quelques comparaisons, qui contiennent, des comparaisons entre les courbes citées au dessus et cela a pour objet de savoir la courbe la moins complexe, celle sélectionnée sera aussi la plus sécurisée.

5.6.1 Infrastructure de l'implémentation

Implémenter un schéma d'IBE repose sur la réussite de l'implémentation d'un couplage, de son tour cette dernière demande une génération d'une courbe et d'un couplage bien adaptés.

Aspects implémentation des couplages bien adaptés

Un couplage bien adapté nécessite de faire attention à deux choses essentielles : Efficacité des couplages, plus précisément réduction de leurs taux de calculs ; attaques MOV [134] et FR [78]. Un couplage est construit de deux bornes, une qui l'initialise, il a comme taille $log_2(r)$. L'autre où vive ses résultats dont $log_2(p^k)$ est le nombre de ses bits, avec p est la caractéristique du corps de base. Pour implémenter un couplage, nous devrons passer par l'algorithme de Miller. Un simple aperçu sur cet algorithme, démontre que nous pouvons nous baser sur l'arithmétique du corps de base F_p.
Les courbes adpatées qui ont un rapport : ρ = $\frac{log(p)}{log(r)}$ proche ou égal à 1, sont plus privilégiées dans le calcul du couplage. Ce rapport mesure la perte d'efficacité de l'arithmétique du corps fini F_p par rapport à la taille de r. Puisque F_p est le corps de base et r / $p^k - 1$ (r $\geq$ p), donc si r est proche de p nous gagnons plus d'efficacité. Mais, il faut faire attention aux niveaux sécurité et à la deuxième condition (attaques de MOV et FR). Le tableau 5.5 s'approche des niveaux demandés [79 pour r, p selon le type de ρ utilisé.

Niveau de sécurité	Taille de r	Taille de p^k	k pour ρ=1	k pour ρ=2
80	160	960-1280	6-8	2-4
128	256	3 000 - 5 000	12 - 20	6 - 10
192	384	8 000 - 10 000	20 - 26	10 - 13
256	512	14 000 - 18 000	28 - 36	14 - 18

TABLE 5.5 – Niveaux de sécurité demandés pour r et p selon le type de ρ utilisé

Nous remarquons alors que pour des niveaux de sécurité équivalents, lorsque ρ=1 le k augmente mais p diminue ce qui rend les calculs plus efficaces. Par contre, pour ρ=2 on a le contraire, on perd alors de l'efficacité.

Étudier l'efficacité d'un couplage a été traité fréquemment par la littérature (en conduisant depuis 2007 à l'engendre d'une conférence annuelle portant le nom des couplages, s'appelle Pairing). Mais on se limite à rappeler ci-après les sept propositions de Galbraith [44] qui permettent d'atteindre une efficacité désirée.

1. Utiliser un r de poids plus petit que possible càd le rendre creux.
2. Travailler le plus possible dans F_p au lieu de F_{p^k}
3. Utiliser une arithmétique efficace pour F_p et F_{p^k}
4. Éviter les calculs lourds comme la division, utiliser tant que possible le carré à la place des multiplications
5. Trouver une méthode qui peut rendre l'exponentiation finale efficace.
6. Noter que si k $>$ 1 et r un nombre premier, r ne divise pas p-1 et r / p^k-1 implique que la $(p-1)^{ième}$ puissance de tous les éléments $F_p{}^*$ est 1. Alors tout terme dans l'algorithme qui appartient à $F_p{}^*$ peut être ignorer
7. Si P $\in$ E(F_p)[r] et Q=(X,Y)$\in$ E(F_{p^k}) avec X $\in F_{p^{\frac{k}{2}}}$. Alors selon [15], le calcul du dénominateur dans l'algorithme de Miller peut être débarrassé.

Pour réussir la proposition numéro 3, nous utilisons des corps, appelés corps adaptés aux couplages.

Aperçu sur les corps adaptés aux calculs des couplages

Un corps adapté au calcul du couplage est celui qui permet du point de vue arithmétique, d'avoir une efficacité dans son calcul. La littérature marque deux travaux essentiels pour les corps adaptés au couplage, l'objectif premier de ces travaux est de comparer les deux couplages Weil et Tate. Le premier travail est celui de Koblitz et Menezes [122], en utilisant des corps de degré de plongement $k = 2^i3^j$ où i,j $\in$ Z, les auteurs démontrent que Weil est plus efficace que Tate. Tandis que le deuxième revient à Granger, Page et Smart [96], ces auteurs se sont basés sur des corps cyclotomiques et ils sont arrivés à un résultat contraire.
Ce débat a été parachevé dans [13] où les auteurs utilisent le corps avec $k = 2^i3^j$ et ils ont trouvé le même résultat [13] que Granger et al. Effectivement, ce résultat semble logique,

d'après les formes finales de Weil et Tate (Weil utilise deux fois Tate). Les corps utilisés dans [122] qui ont un $k = 2^i 3^j$ sont plus généraux que les corps cyclotomiques utilisés dans [96], puisque ces derniers sont restreints à k=6 ou ses multiplicités. Généralement, ces deux corps sont connus par corps amis.
Les corps amis qui ont $k = 2^i 3^j$ ont des représentations très importantes, qui permettent avoir de bonnes réductions.

Théorème 5.6.1 [127] : *Soit $\beta \in F_p$ qui n'est ni un carré ni un cube dans F_p, et F_{p^k} un corps ami qui a un k = $2^i 3^j$. Alors le polynôme $X^k - \beta$ est irréductible dans F_p.*

D'après ce théorème, F_{p^k}=F_p / $X^k - \beta$. Puisque, k = $2^i 3^j$ on aura :

$$F_{p^k} = F_{p^1} = F_p/(X_1{}^{k_1} - \beta_1) \rightarrow F_{p^2} = F_{p^1}[X_1]/(X_3{}^{k_2} - \beta_2) \rightarrow ... \rightarrow$$

$$\rightarrow F_{p^i} = F_{p^{i-1}}[X_{i-1}]/(X_i{}^{k_i} - \beta_i) \rightarrow F_{p^{i+1}} = F_{p^i}[X_i]/(X_{i+1}{}^{k_{i+1}} - \beta_{i+1})$$

Où les β_i ne sont ni des carrés ni des cubiques et les k_i sont les représentations de k, ils peuvent être 2 ou 3. Bien entendu, l'arithmétique avec ces sous-corps se fait par Karatsuba ou Toom Cook selon si 2 ou 3 qui sont utilisés. Ces deux multiplications sont privilégiés pour la raison qu'elles ont des coups de complexités $O(m^{log_2(3)})$ et $O(m^{log_3(5)})$ respectivement par comparaison avec la méthode scolaire. Le corps ami F_{p^k} avec k = $2^i 3^j$ à un autre avantage qui se résume aux simplifications qu'il apporte son polynôme irréductible puisqu'il peut avoir une forme très simple $X^k - \beta$. Nous privilégieons donc l'utilisation de ce type de corps dans l'implémentation des couplages (alors à l'implémentation de l'IBE).

Pour réussir la proposition numéro 5 de Galbraith [44], il faut suivre la méthode qui améliore l'efficacité de l'exponentiation.

Exponentiation efficace pour Tate et ses variantes

Le couplage de Tate et ses variantes (Ate, twist ate..), nécessite le calcul de l'algorithme de Miller plus une exponentiation. Cette dernière dépend de l'exposant $\frac{p^k-1}{r}$ qui est grand puisque nous devrons travailler dans des niveaux de 128-bits et plus pour p et r. Koblitz et Menezes [122] sujettes la divisé en deux parties :

$$\frac{p^k - 1}{r} = \frac{p^k - 1}{\phi_k(p)} \times \frac{\phi_k(p)}{r} \tag{5.69}$$

Avec $\phi_k(p)$ représente l'évaluation en p du k-ième polynôme cyclotomique.
Le $\phi_k(p)$ divise bien p^k-1 puisque ce dernier se factorise à l'aide des polynômes cyclotomiques :

$$p^k - 1 = \prod_{k'/k} \phi_{k'}(p) \tag{5.70}$$

Qui nous permet aussi d'en déduire que r divise $\phi_k(p)$, d'après le fait que k est le plus petit entier tel que r divise $p^k - 1$.
La partie $\frac{p^k-1}{\phi_k(p)}$ est simple, son calcul demande que des calculs par des opérations de Frobenius. Quant à la deuxième $\frac{\phi_k(p)}{r}$ elle reste laborieuse. Scott et ses étudiants [47] [65] proposent d'utiliser l'exponentiation par chaine, plus les vecteurs d'Olive, ceci dépend du degré de plongement k utilisé. Pour plus de détails nous renvoyons aux deux thèses [47] [65].

Quant à la proposition numéro 4, le cas particulier où k est paire permet de faire plus de réductions dans le calcul des couplages. En effet, dans le cas d'un élément b d'ordre r on a :

$$\begin{cases} & b^{r \times \frac{p^{\frac{k}{2}}+1}{r}} = 1 \\ \iff & b^{p^{\frac{k}{2}}+1} = 1 \\ \iff & b^{p^{\frac{k}{2}}} = b^{-1} \end{cases} \tag{5.71}$$

Alors l'inverse des éléments qui ont l'ordre r peut être simplifier au calcul d'une exponentiation. Par conséquent, le calcul du couplage de Weil qui demande le calcul : 2 Miller + Inversion peut se faire par 2 Miller + Exponentiation. Effectivement, ceci est avantageux, puisque l'exponentiation est moins coûteuse que l'inverse (il est montré dans [176] qu'utiliser l'exponentiation pour Weil sera plus efficace).
Ainsi, il est conseillé d'utiliser le plus possible la formule de Fermat (petit théorème de Fermat) : $b^{p-2} = b^{-1}$ mod p. À partir de cette formule, on peut extraire d'autres formes selon le type de p utilisé et selon le corps où les calculs sont développés.
Aussi, nous préconisons de penser aux coordonnées Jacobiennes, avec lesquelles, on n'utilise plus l'inverse dans les étapes Addition et Doublement pour calculer le scalaire kP. Nous prônons aussi d'utiliser l'algorithme de Miller amélioré, qui n'utilise la coûteuse inverse que dans la dernière étape en réduisant le figure de cette opération de deux à une.

La proposition numéro 7 peut être activée en utilisant les courbes tordues et un degré de plongement de parité paire.

Optimisation à l'aide d'une courbe tordue sous un degré de plongement pair

Si k est paire, alors $p^{k/2}-1$ divise $\frac{p^k-1}{r}$. En effet, $p^k-1=(p^{k/2}-1)(p^{k/2}+1)$. Mais, r est premier et r divise p^k-1 implique automatiquement que r divise l'un des deux (d'après Gauss). Le paramètre r ne divise pas $p^{k/2}-1$, puisque, si c'est oui il contredit le fait que k est le plus petit élément tel que r divise p^k-1.

Après avoir utilisé une tordue de degré paire (2,4 et 6), les deux vecteurs v_1 et v_2 qui figurent dans l'algorithme de Miller appartient au corps $F_{p^{k/2}}$. Donc, si on enlève à la puissance $\frac{p^k-1}{r}$, le calcul de ces deux vecteurs sera négligé. Par conséquent, la mise à jours de f_2 et g_2 dans Miller sera aussi négligée (voir [15] pour plus de détails).

En plus, de ces offres d'optimisations, utiliser une courbe tordue dans le calcul d'un couplage permet de travailler dans le corps $F_{p^{k/d}}$ au lieu du corps F_{p^k} où d est le degré de la tordue. Ce qui ramène à faire les calculs dans le sous-corps composé au lieu du corps tout entier.

Pour bien optimiser le calcul d'un couplage, on peut aussi bénéficier des propositions numéro 1, 2 et 3 de Galbraith [44]. Mais, ceci dépend du type de p et k utilisés, à leurs tours, ces deux paramètres dépendent du genre de la courbe qu'ils représentent (adaptée au calcul du couplage). L'optimisation des couplages peut être atteinte par d'autres choses, par exemple, les couplages optimaux (Ate, Twist ate..) sont moins coûteux au niveau itérations que les classiques Tate et Weil. Il est alors conseillé de travailler avec ces nouveaux nés des couplages.

5.6.2 Première implémentation : Mise en évidence du quatrième schéma de l'IBE

• Logiciel utilisé

Comme l'IBE est une technologie plus récente, réussir son implémentation demande plus d'effort. En utilisant le logiciel JAVA, la société Voltage en USA sous la direction de Boneh et ses employés a pris l'initiative d'implémenter le schéma de Boneh et Franklin pour servir ses clients. Scott et ses étudiants préfèrent implémenter et utiliser Sakai Kasahara pour rendre service aux clients de la société Identum qui réside en United Kingdom, Boyen et Martin [32] ont donné un peu de théorie (peu précise) qui peut guider à implémenter BB1 sans préciser aucun logiciel.

Certains praticiens privilègient l'utilisation du logiciel C++ à l'aide de quelques bibliothèques comme : GMP (Gemplus Multiplication Précision), cette bibliothèque est utilisée pour calculer l'arithmétique telle que la multiplication, l'inversion, l'exponentiation etc. Des codes prêts sont présentés dans le site de la bibliothèque [97]. Il y a aussi NTL [163] inventée par Victor Shoup,

cette bibliothèque est privilégiée puisqu'elle offre la possibilité de travailler avec des nombres très grands, chose non vérifiée dans le cas de C++ sans NTL. Ainsi, elle permet de choisir des entiers arbitraires. La bibliothèque NTL est faite pour calculer les polynômes, matrices etc. On peut marier [163] entre NTL et GMP pour bénéficier des services des deux bibliothèques. Pour implémenter le quatrième schéma de l'IBE, nous pensons à l'utilisation du logiciel Magma. Pour la raison que ce dernier suit la progression mathématique et cryptographique (il a traité pas mal de découvertes dans ces deux thématiques), il est flexible à la programmation, il permet de choisir des nombres arbitraires, ainsi, certaines fonctions peuvent être générées par un simple appel à des syntaxes plus simples (appels externes). Le seul inconvenient de ce logiciel est qu'il n'est pas ouvert à un accès libre (la version calculateur est accéssible sur [131]).

• **Préparation et présentation de l'implémentation**

Pour implémenter le quatrième schéma de l'IBE, nous l'avons réalisé dans le niveau bas : 124-bits qui est équivanlent au niveau 64-bits dans le cas de la cryptographie symétrique. Ce niveau est loin des attentes de sécurité pour les schémas cryptographiques où il faut prendre au moins 256-bits pour résister à EPDL (qui est équivalent à 128 dans le cas de la cryptographie symétrique) ; mais il nous permet d'avoir une arithmétique plus simple, qui ne demande aucun appel aux algorithmes spécifiques dans l'accélération de celle-ci. Utiliser le niveau 64-bits est a peu près compatible avec la fonction de hachage SHA2 de 256-bits qui est la sortie la plus basse des fonctions SHA-2 (pour bénéficier de cette fonction, on peut utiliser crypto++ [60]).

Présentation de l'implémentation

Comme nous l'avons signalé dans le chapitre 1 partie 1.4.1, le chiffrement basé sur l'identité repose sur les quatre axes :

Setup : Il demande à l'utilisateur de choisir un type de courbe (supersingulière ou ordinaire) et un degré de sécurité et génèrer ensuite la courbe adaptée. Cet axe est purement responsable de la génération des paramètres publics et la clé maîtresse. Lors de la programmation nous appelons les paramètres générés par cet axe par ***publics.ibe*** et la clé maîtresse par ***clé maîtresse.ibe***.

Extract : Il fournit à partir de l'identité de l'utilisateur et la clé maîtresse, une clé privée. On appele le fichier qu'il généré : ***clé privée.ibe***

Encrypt : Cet algorithme demande à l'utilisateur de saisir un destinataire et un message, puis de lui retourner le message chiffré correspondant. L'axe sera nommé ***encrypt.ibe***

Decrypt : Il décrypte un message chiffré en utilisant la clé privée de l'utilisateur. Ce dernier axe se dénomme ***decrypt.ibe***.

Exemple concret :

Pour simplifier l'implémentation, nous avons pensé à un schéma sous forme CPA. De cette manière, on réduit l'utilisation des fonctions de hachage (non utilisation de H_3 et H_4 dans le quatrième schéma de l'IBE) ainsi que quelques paramètres. Extraire cette forme est simple, elle rassemble le schéma basique de Boneh et Franklin ; mais sous des paramètres différents. Lors de l'implémentation, nous avons rencontré deux problèmes :

1. La bilinéarité des couplages n'est pas assurée par le pratique, cela peut être à cause du code utilisé dans la programmation (pourtant le code utilisé sera donné dans l'annexe-Listing 4, il est est a peu près similaire à celui de Dominguez Perez dans sa thèse 65).
2. Étant donné sP, calculé $s^{-1}sP$ ne donne pas P, mais il donne un autre point (qui est aussi engendré par P).

On a géré ce second problème par la méthode suivante :
Selon une caractéristique p, on a : ss^{-1} = qp+1 et donc calculé ss^{-1}.P revient à calculer (qp+1).P. Alors au lieu de publier P et sP, il est conseillé de publier (qp+1).P et sP. De cette manière, si on calcule ss^{-1}P, on tombe sur (qp+1).P ; pour avoir P il faut que (qp+1) soit l'ordre + 1 de P ou un de ces multiples, ce qui n'est pas toujours vérifié.
Pour gérer le premier problème nous avons pensé à un schéma basique sous la forme :

Schéma basique sous une forme modifiée du quatrième schéma de l'IBE

Setup : Soit (G_1, G_2, G_T) des groupes bilinéaires définis sur un corps de caractéristique p.
Choisir un P dans G_1 et un P_2 dans G_2.
Calculer deux fonctions de hachage :

$$H_1 : \{0,1\}^* \longrightarrow {Z_p}^*,\ H_2 : G_T \longrightarrow \{0,1\}^n$$

L'espace du message est : $\mathbb{M}$ = $\{0,1\}^n$.
L'espace du ciphertext est $\mathbb{C}$ = $G_1^* \times \{0,1\}^n$.
Le système des paramètres publiques est :

$$M_{pk} = \{P, P_2, P_1 = (q_1 p + 1).P \quad où \quad ss^{-1} = q_1 p + 1, aP_2, \hat{e}, G_1, G_2, G_T, H_1, H_2\}$$

La clé *maîtresse* est : **a**=s^2 et **s**.

Encrypt : Étant donné m $\in$ M , ID_B et M_{pk}, faire les étapes suivantes :

1. Choisir arbitrairement un t dans Z_p.
2. Calculer $\hat{e}(t.P_1, (H_1(ID)^2 + a).P_2)$.

   Le ciphertext est alors :

$$CT =< t.s.P, m \bigoplus H_2(\hat{e}(t.P_1, (H_1(ID)^2 + a).P_2)) >$$

Extract : Après avoir reçu CT, Bob envoi t.s.P à la PKG, cette dernière authentifie tout d'abord Bob et elle l'envoi $t.P_1$ = s^{-1}.t.s.P

Nous signalons que la clé privée est bien :

$$\frac{d_B}{H_1(ID_B)} = (\frac{H_1(ID_B)^2+a}{sH_1(ID)}).P_2$$

Pour déchiffrer le message, il suffit d'utiliser d_B.

Decrypt : Étant donné ID_B, d_B et $t.P_1$. Bob peut déchiffrer le message.

Nous proclamons que le couplage ê pris dans ce schéma est une application qui relie uniquement deux points, il a le corps d'un couplage et il est bien calculable.

Méthode de génération des paramètres :

Pour faire cette implémentation nous avons pensé à la courbe MNT qui a le degré de plongement k=3 (pour un D=19).

L'équation de cette courbe est (voir l'annexe pour la description de la méthode de génération de la courbe) :

y^2 = x^3 + *525206902434456156369417200491124332423529878342847678 x +*

488156338200913826313675721737185900932308458793909587

Elle est définie sur le corps :

L=F_p où p =6363585951350831465366416367529396268971941369896619 51

Dans Magma cette courbe se note :

E := EllipticCurve ([L !525206902434456156369417200491124332423529878342847678,

L !488156338200913826313675721737185900932308458793909587]) ;

L'ordre de E est bien :*6363585951350831465366416364929673675407178990290535 75*

Après avoir factorisé cet ordre, on trouve : *r= 3152100205524560584182704364081129757 1*

Par : *i :=Floor(Log(2,3152100205524560584182704364081129757 1)) ; i ;* on obtient : *124*(bits) qui est le niveau de bits de r, il constitue le niveau de sécurité du schéma tout entier. Ce niveau est équivalent à *64* bits dans le cas de la cryptographie symétrique voir chapitre 1 tableau 1.4

(c'est un niveau moyen il est en courte vie, mais il est loin d'être désirable).

Avec : Random(E), on génère le point :

[33697010835573609001801157946891739024829502413722316,

26812752890313846035833632237687589510842806264842381 7 , 1]

On cherche ensuite un groupe G_1, ce groupe est engendré par P et il a comme ordre r.

Le groupe qu'on cherche n'est autre que : $r_1 * P$, il a un ordre r lorsque :

r_1 := *L!(Order(P)/315210020552456058418270436408112975 71);*

ou

r_1 := *L!(Order(E)/315210020552456058418270436408112975 71);*

Aussi avec Random(G) ; on génère le point :

*[2004930613680625148086403383094037363484549617650785 7*K.1*^{2} +

*[5303124420367739089899844111932677812181816053721764 74*K.1 +

13825617230751338373710616176809460933923348299305466 3,

*12318472228324269578379164924047165827275909779878041 8*K.1*^{2} +

*59410839904344032048365994029487282110594205138851170 5*K.1 +

12459532852314619825263442051645518272857440610662813 7 , 1]

L'autre groupe qui est G_2 est généré par :

K!(Order $< P_2 >$ /r)*$< P_2 >$ ou K!(Order (G)/r)*$< P_2 >$.

Avec le fait que :

K :=FiniteField(6363585951350831465366416367529396268971941369896619 51,3);

Et

G := EllipticCurve ([K!5252069024344561563694172004911243324235298783428476 78,

K!4881563382009138263136757217371859009323084587939095 87]);

c'est une courbe elliptique sur le corps fini : F_{p^3}

où p = 6363585951350831465366416367529396268971941369896619 51

Nous avons pensé aussi à l'utilisation du couplage de Weil.

Alors, le premier argument du couplage en utilisant :

*L!(Order(E)/315210020552456058418270436408112975 71)*P;*

Il nous donne : *[25030862124124714019367708100690352468610881476623 6000,*

44840007356869762246244968182713694249356848967628607 2, 1]

Le deuxième argument du couplage se calcule par :

K!(Order(G)/315210020552456058418270436408112975 71)P_2 ;*

Qui nous donne :

*[56478413440898972637817556545185434551951455512005225*K.1*^{2} +

*34468734613869834063477968127629709768501620124646456 2*K.1 +*

595058775441537896100615679176524636221109172121727922,

*27470384327863802622463324097924950634822492063885013 3*K.1^2 +*

*40127158947162932762003481192223503684379299592327837 6*K.1 +*

30227858937845583123418554730178933894326108847346064 5, 1]

Méthode pour calculer q_1 :

Tout s'abord il faut calculer s^{-1}=L!(1/s), ensuite $s^{-1} * s$ et alors : q_1= $s^{-1} * s$ div q.

Pour *s* = *443505922387244651495283053486180215104151970143239 44*

et q= 6363585951350831465366416367529396268971941369896619 51

On a : q_1= *38185492535492880774249727367204124780402063539392665*

Étape *Setup* :

Magma/program-thesis-rkiaaouinatou$publics-ibe

P=[25030862124124714019367708100690352468610881476623600 0,

44840007356869762246244968182713694249356848967628607 2, 1]

P_1*=[19344823433981837652748551919295731963625618425769833,*

50958723885409165767859446379748304707047700264685888 1, 1]

P_2*=[56478413440898972637817556545185434551951455512005225*K.1^2 +*

*34468734613869834063477968127629709768501620124646456 2*K.1 +*

59505877544153789610061567917652463622110917212172792 2,

*27470384327863802622463324097924950634822492063885013 3*K.1^2 +*

*40127158947162932762003481192223503684379299592327837 6*K.1 +*

30227858937845583123418554730178933894326108847346064 5, 1]

aP_2*=[20938440178670310089263286050430748833344363960477003 7*K.*1^2*+*

*49262589864603926018695016896403349687448358640585486 6*K.1 +*

58075839113028460979829495784280507833642042275921063 2 :

*22557903550865991506933090345533706202161917959755739 0*K.*1^2*+*

*17747037744307832464823015587500320064551996600870217 8*K.1 +*

43798876948222149714411128883859154754847010270980844 6 : 1]

Plus, G_1, G_2, SHA_{256} et le couplage de Weil.

Magma/program-thesis-rkiaaouinatou$clémaîtresse-ibe

s = 443505922387244651495283053486180215104151970143239 44

a= 426019378571326201576786590890126082493938061359166192

Étape *Encrypt* :

Magma/program-thesis-rkiaaouinatou$encrypt-ibe

Le random t (aléa) qu'on a choisit est :

t= 405079451177315608599513757320168911191961687767805270

Alors, t*s*P est :

CT_1*=[172516063635990923196671854451607694170438473732217530,*

129894732382138464724693976102635687643201563948578625, 1]

L identité choisie est : ID_B = rkiaaouinatou@yahoo.fr.

La taille du message par bloc est : *n = 64*.

Le message (m) choisi est : ≺ *Rapport de thèse de Rkia Aouinatou* ≻.

On a : ($H_{256}(rkiaaouinatou@yahoo.fr)^2$+a)*$P_2$ =

*[24853210010018459446098314907322471300648722366366 4868*K.*1^2+

*22157367842411174276575795094576378794467979830864 8289*K.1 +*

50007883934970569924548120857874842090009752810543 3142,

*14515774937039334183909439631079741536867493287102 8819*K.*1^2+*

*54937910085421043622116688966702235373053323607490 8280*K.1+*

76126289321517652652869184710984483183722644927303226, 1]

Ainsi, nous avons pour :

r := 315210020552456058418270436408112975 71 ;

P :=E![33952939344260725317400693320215820042543646300182423 3 ,

33781602964613168858399826480444129946691527598868606, 1] ;

*Q :=G![24853210010018459446098314907322471300648722366366 4868*K.*1^2*

*+22157367842411174276575795094576378794467979830864 8289*K.1 +*

50007883934970569924548120857874842090009752810543 3142 ,

*14515774937039334183909439631079741536867493287102 8819*K.*1^2 +*

*54937910085421043622116688966702235373053323607490 8280*K.1*

+76126289321517652652869184710984483183722644927303226, 1] ;

Le calcul de : Miller(P,Q,r)/Miller(Q,P,r) ; (couplage de Weil : $e(t * P_1, (H_1(ID)^2 + a) * P_2)$), nous donne comme résultat :

*246705628412653617049239734310197585564139077763982898*K.*1^2 +

*352478617924475877022054753694220732446924902619701031*K.1* +

382946293072451801350540092797687815750627712337607982

Après, on lui applique une fonction de hachage de type 256 (H_{256}), on a :

H_{256}(*246705628412653617049239734310197585564139077763982898*K.*1^2 +

*352478617924475877022054753694220732446924902619701031*K.1* +

382946293072451801350540092797687815750627712337607982)

= *4c1f101e1c8bb0fbf38bd0b6dedb0abf0f73aecc8b85fbc23336c48554f61f4e*

Après avoir convertir, découper le message par des blocs de 64-bits et faire l'opération XOR. Le message chiffré (en base 64) sera alors :

< *CT2* = *STkHGRp8l7n1BO7wwrYYosySGoitmA3XSlJsaBQgFJ2O4fTv9Q* = = >

Étape *Extract* :

Magma/program-thesis-rkiaaouinatou$cléprivée-ibe

Après avoir que Bob reçu :

[172516063635990923196671854451607694170438473732217530,

129894732382138464724693976102635687643201563948578625, 1],

il demande le calcul de : $s^{-1} * t * s * P$.

A noter que la clé privée est :

d_B = *[248532100100184594460983149073224713006487223663664868*K.*1^2

+ *221573678424111742765757950945763787944679798308648289*K.1* +

500078839349705699245481208578748420900097528105433142 ,

*145157749370393341839094396310797415368674932871028819*K.*1^2 +

*549379100854210436221166889667022353730533236074908280*K.1*

+761262893215176526528691847109844831837226449273032 26, 1]

Étape *Decrypt* :

Magma/program-thesis-rkiaaouinatou$decrypt-ibe

Après qu'il reçoit les paramètres de l'étape Extract, Bob déchiffre le message.

5.6.3 Deuxième implémentation : Courbe et schéma les plus convoités

• Courbe la plus désirée

Les courbes citées dans la partie 5.5.2 permettent de résister à l'attaque présentée dans la section 5.4. Il est ordinaire que les courbes de Barreto-Naehrig sont les plus préférables parmi les courbes adaptées au calcul des couplages, puisqu'elles sont flexibles pour une utilisation d'un tordu de degré 6 qui est le plus grand degré qui existe d'un Twist, elles ont un degré de plongement sous forme $k = 2^2 \times 3$ qui est bien compatible au corps adapté au calcul d'un couplage, ainsi, leurs polynômes cyclotomiques ($\phi_{12}(p)$ où p est la caractéristique du corps de base) à un grand degré. Afin de savoir des idées sur les caractéristiques des autres courbes, on se refère à l'étude suivante :

Tout d'abord, nous lançons un coup d'oeil sur la flexibilité des courbes de la partie 5.5.2 avec l'attaque 5.4. Pour résister à cette attaque, les courbes de Brezing-Weng et Cocks-Pinch peuvent nous servir uniquement dans le cas où la condition numéro 1 citée dans la partie 5.4.4 soit vérifiée, c'est à dire dans le cas des couplages asymétriques, il nous défend d'utiliser les couplages symétriques. Mais, nous signalons que dans ce cas l'attaque préexiste encore puisque les conditions 2 et 3 de la partie 5.4.4 restent à la porté. Par ailleurs, nous avons cité les courbes MNT parmi les courbes qui peuvent résister à l'attaque 5.4, d'après le fait que ce genre de courbes peut avoir un ordre impair ; mais en réalité cette condition n'est pas toujours vérifiée. Par contre, les courbes de Barreto-Naehrig [17] et celles de Freeman[80] permettent de bien résister à l'attaque 5.4 (on a bien r=p+t-1 est un nombre premier, le second terme est l'odre de la courbe).

Autres caractéristiques des courbes de Barreto-Naehrig, Freeman (courbes qui résistent bien à l'attaque présentée dans la partie 5.4.4) et celles de MNT, Brezin Weng et Cocks Pinch(courbes qu'on peut réctifier) sont citées dans les tableaux 5.6, 5.7 et 5.8.

Dans le tableau 5.6, nous rappelons les propriétés courantes de ces courbes.

Courbes/Propriétés	ρ	degré de plongement k	twist convenable
MNT[139]	1 pas toujours	3, 4 et 6	d=2,3 et 6
Barreto-Naehrig[17]	1	12	d=3, 2, 4 et 6
Freeman[80]	1	10	d=2
Brezing-Weng[43]	$1 \leq \rho \leq 2$	n'importe	dépend de k
Cocks-Pinch[59]	2	n'importe	dépend de k

TABLE 5.6 – Quelques caractéristiques des courbes citées dans la partie 5.5.2

Dans un premier temps nous allons approcher le lecteur sur les difficultés rencontrées lors de la génération des courbes citées (cette remarque sera aussi valable pour le survey des

courbes adaptées au calcul des couplages donné par Freeman et al dans le papier [79]). Nous avons remarqué que toutes les méthodes qui permettent de générer les paramètres : p, t et r (paramètres de la courbe) sous forme de polynôme, sont faciles d'être créer puisqu'on pu uniquement donner des x qui engendrent p(x) et r(x) premiers. Sous le logiciel Magma, il suffit d'appeler à l'ensemlble : {**x : x in [**interval de définition**]** / **IsPrime**(formule de p(x)) **and IsPrime**(formule de r(x))}. Les courbes de Barreto-Naehrig, Freeman et celles de Brezing-Weng sont parmi cette catégorie. Nous notons que les courbes de Brezing-Weng sont incommodées à cause des conditions demandées lors de la génération des corps K qui doivent être sous la forme Q(x)/ r(x) et qui doivent aussi contenir $\sqrt{-D}$. La même chose sera dite sur les courbes de Cocks Pinch ; mais, avec ces dernières le problème est moins aigu puisqu'il faut penser à un r premier et qui vérifie k|r-1 et $(\frac{-D}{r})$=1, ceci peut être géré sous Magma. Par contre, les courbes de Barreto-Naehrig et celles de Freeman ne présentent aucun obstacle, il suffit de trouver le x convenable. Les courbes de Barreto-Naehrig sont plus privilégiées, puisqu'elles permettent d'être générées par un simple algorithme (donné dans l'annexe-Listing 1, voir aussi le papier [17]) au lieu de parfaire les étapes (• et ••) de la partie 5.5.1.

En contrepartie, la création des courbes MNT est étroite, du fait que les nombres p qui satisfont la formule : $(x_p + \sqrt{3D}y_p)(x_0 + \sqrt{3D}y_0)^k$ sont très rares et ceci conduit à ne pas être libre dans le domaine du travail (le niveau de sécurité du bit p), la méthode de Cocks Pinch présente le même problème. Nous remarquons aussi que les courbes produites de la formule $y^2 = x^3 + 3k_1c^2x + 2k_1c^3$ sont plus complexes, tandis que les courbes de Barreto-Naehrig (celles qu'on génère par l'algorithme présenté dans [17], donné aussi comme Listing 1 dans l'annexe) sont très simples, puisqu'on a : le a=0 (en considérant $y^2 = x^3 + ax + b$ comme équation d'une courbe elliptique) ainsi le b est très petit.

Courbes/Propriétés	Construction	Domaine de travail	Forme de la courbe
MNT[139]	trop d'étapes	très rare & vague	très complexe (dans des niveaux grands)
Barreto-Naehrig[17]	plus que facile	vaste	Plus que simple quel que soit le niveau
Freeman[80]	facile & peu d'étapes	vaste	très complexe (dans des niveaux grands)
Brezing-Weng[43]	très difficile	un peu vaste	très complexe (dans des niveaux grands)
Cocks-Pinch[59]	un peu difficile	peu (rare & vague)	très complexe (dans des niveaux grands)

TABLE 5.7 – Comparaison entre les courbes de la partie 5.5.2 selon quelques points

On peut dépouiller d'autres remarques qui concernent le niveau qu'offre chacune des courbes selon l'intérêt du degré de plongement qui les caractérisent. Avant de citer nos remarques,

nous proposons de fixer le degré de plongement k pour les courbes qui fonctionnent avec des k quelconques. En ce qui concerne Cocks Pinch, nous proposons de choisir k=5, ce dernier est premier (alors il vérifie la condition 1 de la partie 5.4.4) et il est plus petit que les degrés de plongement qui caractérisent les autres courbes qui sont citées dans la partie 5.5.2, on a choisi k de telle caractéristique à cause du rapport de performance ρ de Cocks Pinch qui est égal à 2. Suivant la même raison, nous choisissons un k=7 pour Brezing-Weng puisque son $1 < \rho \leq 2$. Pour MNT nous préférons travailler avec k=6 qui est le plus grand degré de plongement qui existe pour les courbes MNT, ainsi, il est proche aux degrés de plongement utilisés par les autres courbes.
En se basant sur l'égalité $p^k - 1 = \prod_{d/k} \phi_d(p)$, nous pouvons extraire facilement le suivant :

k=12, $p^k - 1 = (p-1) \times (p+1) \times (p^2+p+1) \times (p^2+1) \times (p^2-p+1) \times (p^4-p^2+1)$.
k=10, $p^k - 1 = (p-1) \times (p+1) \times (p^4+p^3+p^2+p+1) \times (p^4-p^3+p^2-p+1)$.
k=6, $p^k - 1 = (p-1) \times (p+1) \times (p^2+p+1) \times (p^2-p+1)$.
k=7, $p^k - 1 = (p-1) \times (p^6+p^5+p^4+p^3+p^2+p+1)$.
k=5, $p^k - 1 = (p-1) \times (p^4+p^3+p^2+p+1)$.

On a r / $p^k - 1$ donc r / $\phi_d(p)$ pour tout d / k et alors le degré de r peut être petit ou égal au degré de $\phi_d(p)$ (qui soit irréductible). Dans les cas ambitionnés, le niveau des r qu'on puisse atteindre est peut être égal au niveau de 2 $\times$ bit de p pour : Barreto-Naehrig, Freeman, Cocks-Pinch (k=5) et Brezing-Weng (k=7) ; 1 $\times$ bit de p pour MNT (k=6). Mais, il faut faire attention au rapport de performance de chacune de ces courbes.
Cela d'une part, d'autre part, on a signalé que réduire l'exposant $\frac{p^k-1}{r}$ peut se faire par la méthode de Kilbotz et Menezes [122] : $\frac{p^k-1}{\phi_k} \times \frac{\phi_k}{r}$ (pour k=12, on a : $\frac{p^k-1}{\phi_k} \times \frac{\phi_k}{r} = (p^8+p^6-p^2-1) \times (\frac{p^4-p+1}{r})$, le bit de cet exposant où $\rho = 1$ est : $\cong$ (Bit de p) $\times$ 32). En utilisant ceci, on peut facilement extraire les courbes (suivant leurs degrés de plongement) qui peuvent nous conduire à des exposants plus réduits. Le tableau 5.8 résume toutes ces remarques dans le cas d'un niveau de sécurité égal à 80 (cas de la cryptographie symétrique).

Courbes/Propriétés	Niveau de r dans le cas ambitionné	Bit de p	$\cong$ Bit de l'exposant pour le niveau 80-bits
MNT[139]	160	160(ρ =1)	160 $\times$ 8 = 1280
Barreto-Naehrig[17]	320	160(ρ =1)	160 $\times$ 32 = 5120
Freeman[80]	320	160(ρ =1)	160 $\times$ 24 = 3840
Brezing-Weng[43]	425,6	212,8 (ρ =1,33)	212,8 $\times$ 6 $\times$ 1,33= 1698,14
Cocks-Pinch[59]	640	320(ρ =2)	320 $\times$ 4 $\times$ 2 = 2560

TABLE 5.8 – Bits de r dans le cas ambitionné et de l'exposant selon un même niveau

Le cas ambitionné signifie le cas le plus possible à atteindre, à titre d'exemple, pour le niveau

80-bits (cryptographie symétrique) et dans le cas des courbes de Barreto-Naehrig[17], on a : ce niveau est équivalent à 160-bits pour p ($\rho = 1$), on peut atteindre alors : 2 $\times$ 160=320 pour r.

Remarque 5.6.1 : *Normalement, les polynômes cyclotomiques ϕ ne sont pas en général irréductibles sur les corps finis. Alors le niveau de r dans le cas ambitionné, n'est pas toujours vrai, pour avoir des résultats exactes, il faut que le $\phi_k(p)$ soit irréductible et vérifie les conditions de la définition 5.5.1*

Nous remarquons qu'avec les courbes de Barreto-Naehrig et Freeman, on peut atteindre des niveaux de r bien souhaités, mais on confronte des niveaux plus élevés dans la puissance par comparaison avec les courbes de Cocks Pinch (pour k=5), Brezing-Weng (k=7) et MNT.
Ce problème peut être réglé en utilisant les twists, on obtient ainsi :

Courbes/Propriétés	Degré du Twist utilisé	Bit de q	Corps F_{p^k}	$\cong$ Bit de l'exposant pour le niveau 80-bits
MNT[139]	6	160(ρ =1)	F_p	160
Barreto-Naehrig[17]	6	160(ρ =1)	F_{p^2}	160
Freeman[80]	2	160(ρ =1)	F_{p^5}	640
Brezing-Weng[43]	-	212,8(ρ =1,33)	F_{q^7}	1698,14
Cocks-Pinch[59]	-	320(ρ =2)	F_{p^5}	2560

TABLE 5.9 – Bit de l'exposant et corps utilisé après l'utilisation des twists convenables

Après avoir utilisé les twists convenables, le k se réduit de : 12 à 2, de 6 à 2 (ou 1) et de 10 à 5 respectivement pour les courbes de Barreto-Naehrig, MNT et Freeman. Pour les deux premières courbes on peut réutiliser un autre twist de degré 2 et on tombera sur le corps simple F_p. Mais pour chaque changement, il faut faire attention aux courbes utilisées et de les changer à la manière suivante :

$$y^2 = x^3 + a'x + b' \quad avec \quad a' = v^{\frac{4}{d}}a \quad et \quad b' = v^{\frac{6}{d}}b \tag{5.72}$$

Où v est la racine de degré exactement d (non moins) dans F_p

Comme guise de conclusion, les courbes de Barreto-Naehrig sont les échéantes d'être considérées dans les implémentations de la cryptographie basée sur les couplages, en particulier l'IBE.

• Schéma plus approprié

Nous l'avons signalé, le schéma de Boneh et Franklin souffre du problème de la fonction de hachage utilisée pour générer la clé privée, puisque, trouver la fonction de hachage d'une

identité qui vérifie : $H(ID)^3 + aH(ID) + b$ sous forme d'un carré, n'est pas toujours offert (pour construire un point de la courbe elliptique). Alors, soit que nous augmentons la projection de H(ID) jusqu'à ce qu'il vérifie la condition recherchée, mais dans ce cas on perd le fait que Q_{ID}(=H(ID)) caractérise la clé privée de ID (qui est égale à sQ_{ID}), ou bien, donner des informations supplémentaires à l'émetteur et au récepteur, encore, dans ce cas on déséquilibre le principe de l'IBE.

Dans les deux cas, ce n'est pas possible de faire cette projection excepté si on a de la chance pour certain ID. C'est pourquoi, le schéma de Boneh et Franklin n'est pas tellement efficace et nous recommandons d'utiliser celui de Sakai-Kasahara. Il faut noter que ce schéma endure, également, d'une faiblesse à laquelle personne n'a fait attention. En effet, le schéma est basé sur le fait que (s+H(ID))$(s + H(ID))^{-1}$.P=P où s est la clé maîtresse et P est un point publié dans le Params, dans le cas d'un corps fini de caractéristique p cela n'est pas vrai puisque (s+H(ID))$(s + H(ID))^{-1}$ donne qp+1. Alors, on ne peut obtenir (qp+1).P=P que dans le cas où qp+1 est l'ordre de P ou de un de ces multiples. Une solution qui peut nous sauver consiste à publier (qp+1).P comme étant point publique au lieu de P, mais nous constatons que ceci n'est pas possible puisque (qp+1) représente l'inverse des ID, varie selon le changement de ces ID. Par conséquent, il faut noter la non efficacité de ce schéma (la même chose sera dite pour BB2 et Gentry qui sont basés sur ce principe).

Dans les chapitres 2 et 3, nous avons cité deux directions des schémas d'IBE : une qui est fondée sur la théorie des nombres (comme RSA, section 2.2 et Cocks exposé dans 2.1), elle est exposé dans le chapitre 2. Tandis que l'autre est basée sur les couplages, nous l'avons cité dans le chapitre 3. Afin de peser ces deux directions l'une par rapport à l'autre, nous ne sélectionnons que les schémas puissants qui représentent chaque direction.

Pour la direction couplage nous éloignons les deux schémas, Boneh-Franklin et celui de Sakai-Kasahara d'après ce qu'on a annoncé plus haut. Aussi, nous écartons le schéma BB1 puisqu'il est plus complexe et nous contentons de sélectionner le nouveau IBE schéma présenté dans la section 3.1 d'après le fait qu'il fonctionne avec un inverse simple (on peut appliquer le principe qui permet de publier (qp+1).P parmi le params avec s^{-1}sP=(qp+1).P), il ne projette pas sur une courbe elliptique, ainsi, il est moins complexe. Pour l'autre direction il est évident de choisir le schéma de RSA (troisième version qui est citée dans la partie 2.2.8).

Le nouveau IBE schéma repose sur le fait de calculer à chaque fois l'inverse de H(ID) qui est un peu coûteux ; mais avec l'utilisation des algorithmes dédiés à cela comme celui d'Euclide Étendu les choses vont marcher. Aussi, le schéma s'appuie sur l'utilisation des couplages qui consomment beaucoup comme nous avons vu durant ce chapitre. De l'autre côté, le RSA souffre du problème de sélection des nombres premiers, ce problème peut être surmonté en

utilisant des techniques et des tests comme Miller-Rabin [89][150] et AKS [1]. Le RSA renferme sur une autre faiblesse qui est le calcul d'une exponentiation modulaire dans le cas des paramètres de tailles plus grandes (exposant et module).

Dans un premier pas, nous proposons de comparer le calcul d'un couplage (couplage de Ate défini sur les courbes de Barreto-Naehrig) figuré dans le nouveau IBE schéma et l'exponentiation modulaire (algorithme Square and Multiply) utilisé par RSA (section 2.2).

	Couplage de Ate	Exponentiation Modulaire
Paramètres	Twist de coordonnées [0,b×v]=[0,5×v]= [0,4138910128900522472684406 3065289848796595752895 48] Corps F_{p^2} Bit de p est **165** & Bit de r : **36**	1 Bloque du message : **17** Bit du module n : **480** Bit de l'exposant e_{ID} : **360**
Temps	**17 ms**	**16 ms**
Espace mémoire	**10 MB**	**10 MB**
Seed	**1499966120**	**2661317510**

TABLE 5.10 – Comapraison entre le couplage de Ate et l'Exponentiation Modulaire sous des niveaux équivalents

Les mesures du tableau 5.10 sont pris en utilisant la version V2.19-4 (version calculateur de Magma) et Intel(R) Core(TM)2CPU 2.13GHz 0.99 Go de RAM et des niveaux équivalents (36 est équivalent à 480 voir tableau 1.4 du chapitre 1).

Les mesures citées dans le tableau 5.10 caractérisent les calculs dans le niveau Encrypt. Concernant le niveau Extract, dans le nouveau IBE schéma il figure le calcul de l'inverse de r_{ID}, pour faire cette opération nous avons pensé à l'algorithme d'Euclide Étendue. Aussi, pour le RSA nous avons besoin de sélectionner à chaque fois de deux nombres premiers, normalement pour le faire, nous devrons programmer un algorithme qui collecte la sélection du nombre et son test. Mais, sous Magma, nous avons pensé uniquement à l'expression :

{p : p in [$2^{bitcible}$..$2^{bitcible}$+domaine cherché]|IsPrime(p)} ;

En utilisant les mêmes conditions du tableau 5.10, on compare dans le tableau 5.11 entre le calcul d'un inverse pour le niveau 160-bits (niveau de q utilisé dans l'expérience précédente) et la sélection d'un nombre premier en utilisant le niveau 120-bits (120×4=480 dans le cas d'un RSA multiprimes).

	Calcul de l'inverse par Euclide Étendue	Sélection des nombres premiers
Temps	**16 ms**	**19 ms**
Espace mémoire	**10 MB**	**12,31 MB**

TABLE 5.11 – Comparaison entre le calcul d'un inverse sous le niveau 160-bits et la sélection d'un nombre premier sous le niveau 120-bits

Il est alors clair que la sélection des nombres premiers utilisés dans le RSA-IBE coûte cher. Mais le RSA qui domine actuellement le marché à d'autres avantages :

— Jusqu'a nos jours, aucune attaque n'a réussi de mettre en défaut le schéma RSA.

— Le RSA est presque sûr et flexible à plusieurs utilisations (se trouve dans de nombreuses applications commerciales).

— Dispose d'une confiance remarquable dans l'entourage scientifique.

Pour avoir des idées sur les attaques éventuelles de RSA, nous renvoyons le lecteur à [167].

5.7 Conclusion & Interprétations

Le dévelleppement de ce chapitre nous a permis de constater que le couplage de Tate (et ses variantes) sert bien la cryptanalyse, puisque :

— Avec le couplage de Tate, la projection du logarithme discret des courbes elliptiques sur les corps finis est moins rigide que celui de Weil.

— La résolution de CDHP et BDH qui sont le coeur de la plupart des cryptosystèmes cryptographiques (par exemple les schémas présentés lors du chapitre 3) est peut être trivial si on ne prend pas quelques précautions.

Pour réussir la dernière condition, nous avons besoin de :

1. Couplages symétriques qui fonctionnent avec des difféomophisms linéaires.

2. L'ordre du premier paramètre du couplage doit être pair.

3. Le #E (cardinal de la courbe elliptique) doit être composé.

Afin de bloquer une telle attaque, nous recommandons d'utiliser plus que possible les courbes ordinaires avec un degré de plongement impair, ce qui permet d'éloigner tous les genres des twists. Nous privilégions alors utiliser des degrés de plongement 11, 13, 17, 19,... ; mais puisque nous avons besoin des twists pour accélérer les calculs des couplages, on peut alors bloquer l'attaque en passant à la condition numéro 2, en fait, il est naturel d'utiliser un ordre premier d'après l'attaque de Pollard ρ, dans la partie 5.4.4 on a montré qu'il est non seulement naturel ; mais aussi nécessaire. En plus, on a démontré que cette condition est insuffisante et on lui ajoute la condition 3 telle que #E=r doit être premier.
Réussir ces conditions est fortement lié à la construction des courbes adaptées au calcul d'un couplage. Dans ce sens, nous recommandons d'utiliser les courbes de Barreto-Naehrig et Freeman. Celles de Brezing Weng et Cocks Pinch résistent à la condition 1 ; mais elles ne

permettent pas de défendre l'attaque puisqu'elles sont susceptibles devant les conditions 2 et 3. Par contre, ces deux courbes sont plus demandées pour réussir Pairing Inversion.
En plus que les courbes de Barreto-Naehrig sont convoitées pour défendre cette attaque, elles ont beaucoup d'avantages, nous les avons rappelé dans la partie 5.6.3. La remarque la plus importante parmi eux est qu'elles sont moins complexes, c'est pourquoi nous les avons considérés dans nos comparaisons entre les schémas des IBE. En utilisant ces courbes ainsi que d'autres paramètres qui rendent le couplage moins complexe (les twists et le couplage de Ate), nous concluons que les schémas qui sont basés sur les couplages sont plus lourds que le RSA proposé dans le chapitre 2. En effet, à part que l'exponentiation est plus rapide que le calcul d'un couplage, les schémas basés sur ce dernier demandent plus de calculs : beaucoup d'opérations arithmétiques (inverse, exponentiation, multiplication scalaire et normal) ainsi que des fonctions de hachage. Toutes ces choses nous conduisent de déduire que le RSA avec l'IBE est le plus dominant.

Cela d'une part, d'autre part, les schémas qui fonctionnent avec les couplages souffrent de plusieurs faiblesses, à part que c'est difficile d'implémenter les notions couplages, où personne ne la réussi. Les schémas qui utilisent ces notions endurent de quelques asthénies. En effet, les schémas sous la catégorie Inverse approche (voir chapitre 1) comme Sakai-Kasahara, Gentry et BB2 sont basés sur l'égalité : (s+H(ID))$(s + H(ID))^{-1}$.P=P ; mais en réalité celle ci donne (s+H(ID))$(s + H(ID))^{-1}$=(q.p+1) et non pas 1. Alors pour que (q.p+1).P = P il faut que (q.p+1) soit l'ordre de P ou un de ces multiples, le problème ne peut pas être résolu puisque H(ID) varie. Il est connu que le schéma de Boneh et Franklin demande une projection sur la courbe elliptique, la solution pratique la plus proche à cette faiblesse est de faire associer à H(ID) le paramètre x et d'augmenter ainsi ce x jusqu'à ce que $x^3 + ax + b$ soit un carré. Le problème se présente lorsque les deux condensés $H(ID_A)$ et $H(ID_B)$ sont confondus, ce qui implique que Q_{ID_A} et Q_{ID_B} s'identiquent aussi, et on aura d_{ID_A} = sQ_{ID_A} = sQ_{ID_B} = d_{ID_B}. Le schéma qui reste efficace parmi les fondateurs schémas déclarés dans le chapitre 1 est celui de BB1, nous proposons une comparaison concrète au niveau complexité entre ce schéma et le nôtre qui est proposé dans la section 3.1 (nouveau IBE schéma) puisque les deux sont dans l'approche des *Commutatives Blinding*, modèle sélective ID ainsi ils sont sous forme CPA (pour BB1 nous prenons la version CPA donnée dans [23]).
Pour faire cette comparaison nous nous sommes basés sur les paramètres suivants :
Une courbe de Barreto-Neihrig, qui a comme paramètres : $y^2 = x^3$ + 11 sur le corps
GF(135276052767772793926434531331917646431033007651947962528424729144584670067070 77202281550480406028079726862634397 3)

Avec cette courbe le degré de sécurité utilisé est 255, qui est le degré le plus souhaité actuellement en cryptographie (chapitre 1). Les paramètres utilisés (α,...) ont des degrés de sécurité entre 360-bits et 366-bits. Pour rester neutre et être correct dans le jugement nous tenons en compte aussi Seed, qui signifie (vitesse de débit). On a alors :

	Schéma BB1	***Nouveau IBE schéma***
Params	Temps : **41** (ms) Seed : 203779131	Temps : **41** (ms) Seed : 738538377
Extract	Temps : **68** (ms) Seed : 4040816338	Temps : **32** (ms) Seed : 2545845857
Encrypt	Temps : **20** (ms) Seed : 788923756	Temps : **20** (ms) Seed : 3938623730
Decrypt	Temps : **61** (ms) Seed : 1298479263	Temps : **39** (ms) Seed : 389952464

TABLE 5.12 – Comparaison entre BB1 et le nouveau IBE schéma dans les quatre étapes : Params, Extract, Encrypt et Decrypt.

Conclusion générale

Résumé des travaux construisant le corps de cette thèse

L'IBE est une technologie permettant de présenter une clé publique sous forme d'une identité. Ce principe la rend plus intéressante que le Standard PKI, puisque, comme il a été exposé dans le chapitre 1 une clé publique sous cadre PKI provoque beaucoup de problèmes. Cette thèse a pour objet d'encadrer deux aspects essentiels pour l'IBE : sa théorie et sa pratique. Les chapitres 2 et 3 sont réservés à la partie théorique et nous avons réalisé en ce sens, ce qui suit :

1. *Apporter un anonymat efficace pour le schéma de Cocks (section 2.1), bien entendu cette propriété permet de bloquer l'adversaire dans un état initial, puisqu'il sera incapable d'extraire l'identité mise en jeu. Du point de vue justice et comme on a exposé dans la conclusion du chapitre 2, notre proposition est plus efficace que celle de Guiseppe Anteniese et Paolo Gasti proposée lors de la conférence CT-RSA 2009, même si on n'a pas encore prouvé sa sécurité.*

2. *Résoudre le restriction du fonctionnement de l'IBE avec les schémas existants. Plus particulièrement, nous avons examiné l'intégration du RSA dans l'IBE. Ce problème qui parait aigu depuis 30 ans (càd depuis 1984) trouve sa résolution dans la section 2.2.*
   Nous avons commencé par développer l'ancienne proposition de Boneh et al lors d'un projet de DARPA et dont ils ont utilisé des techniques typiques comme les SEM, nous avons amélioré leurs propositions de telle manière : qu'elle a moins d'authentifications, ne souffre plus du problème de Key Escrow, avec le cryptosystème construit on peut signer et crypter en même temps. Ainsi, nous avons proposé un exposant convenable pour un RSA avec l'IBE, cette proposition répond bien au challenge de Xuhua Ding et Gene Tsudik qui visent avoir un exposant sans Random Oracle. Toutes ces choses sont couronnées par une nouvelle méthode où on peut intégrer le RSA dans l'IBE en conservant le RSA classique (sans l'aide des SEM).

Ces deux points examinés s'inscrivent dans le cadre des schémas sans couplages. Pour ceux qui l'utilisent, normalement, proposer de nouveaux schémas avec couplage est un peu difficile puisqu'il faut faire attention au niveau Extract et à la manière de présenter la clé privée. Toutes les propositions possibles sont déjà traités par la littérature. Elles sont classées par Boyen selon trois catégories : Full Hash, Inverse approche et Commutative Blinding.

Alors, proposer un nouveau schéma demande soit, de combiner entre ces trois catégories ou de développer les schémas qui les aient satisfaits. C'est pourquoi, nous préférons et nous avons pris l'initiative, de faire une simple classification ; mais aussi efficace des schémas existants de l'IBE, publiée dans le WOTIC11. Le but est de montrer les bienfaits et les inconvénients de chacun d'eux et les schémas qui méritent une standardisation (déja initialisé par IEEE, l'IBE a pris la norme IEEE 1363.3 [107]). Cette classification est plus générale que celle proposée par Boyen dans [23] ; malheureusement nous ne l'avons pas citée dans la présente thèse.

La dite classification, nous permet d'extraire quelques points sur lesquels on peut travailler. Par exemple, nous avons remarqué que le schéma BB1 est plus complexe et que BB2 n'est pas bien classé au niveau sécurité. On en déduit que les schémas sous Random Oracle (Boneh-Franklin et Sakai-Kasahara) souffrent de quelques faiblesses dans le même niveau. Se basant sur cela et sur d'autres points, nous avons apporté dans le chapitre 3 les innovations suivantes :

3. *Dans la section 3.1, nous avons développé le schéma BB1 de Boneh et Boyen en le rendant moins complexe. Ce nouveau schéma s'inscrit dans la classe des Commutative Blinding. On a ensuite prouvé la sécurité de notre schéma dans le modèle sélective ID.*

4. *Par ailleurs, nous avons converti dans la section 3.2, le schéma développé (le même que celui de la section 3.1) à un autre qui supporte HIBE, nommé nouveau HIBE schéma. Une comparaison entre le nouveau HIBE schéma et celui de BBG, permet de déduire qu'avec le nôtre on garde la propriété sélective ID^+ proposée par Sanjit et al [156]. De plus, la projection de la clé privée de notre schéma est dans Z_p au lieu de $Z_p{}^*$ utilisé par le schéma BBG [30]. Ces deux propriétés se sont avérées simples ; mais elles jouent un rôle assez important, puisque, avec la première on garde plus de finesse dans les preuves de sécurité, tandis que la deuxième nous permettra de faire un choix des identités plus vague. En plus de tout cela, on a montré la preuve de sécurité de notre schéma sous un problème plus efficace que celui utilisé par BBG.*

5. *Au final, nous avons présenté dans la section 3.3 un quatrième schéma sous le modèle Random Oracle pour la technologie IBE. Ce schéma ne satisfait aucune des catégories déclarées ci-dessus (proposée par Boyen). En plus, il ne souffre pas des faiblesses dont souffrent les anciens schémas de l'IBE sous Random Oracle (Sakai-Kasahara et Boneh-Franklin).*

Aussi, nous sommes contenté dans le chapitre 4 d'examiner quelques points importants pour l'IBE. Ce chapitre est considéré sur le plan théorique comme un complément des deux chapitres 2 et 3. Selon cette vision, on a mis l'accent sur :

- *Par construction, l'IBE souffre du problème de Key Escrow. Malheureusement, les premières propositions pour résoudre cette corvée renferment un problème plus sérieux et connu par DOD (Daniel of Decryption), càd défendre les utilisateurs de lire leurs messages.*
  Une solution à ce problème a été proposé par Joseph K. Liu et al ; mais elle reste non satisfaisante puisqu'elle ne garantie pas la chose visée.
  Nous avons exposé dans la section 4.2 une méthode où on peut gérer les deux problèmes, dans notre approche on a utilisé le protocole de Diffie-Hellman. Ce choix est flexible aux utilisations multiples : on peut l'utiliser dans des schémas avec ou sans couplage, il peut couvrir la souffrance de l'IBE avec le problème du stockage de la clé maîtresse et celui de la révocation des clés.

- *On a aussi encadré dans la section 4.4, l'IBE avec les attaques à canaux cachés. Dans un premier temps, nous avons proposé une attaque DPA face au couplage, nous avons ainsi traduit cette attaque sur l'IBC en général, enfin, nous avons examiné et ensuite proposé des contres mesures convenables pour toute sorte d'attaque DPA face au couplage. Dans la section 4.5 nous avons proposé une contre mesure idoine pour défendre l'attaque DFA dans le cas où le degré de plongement est pair.*

Dans le dernier chapitre, nous avons traité la partie pratique de notre thèse, au début, nous avons développé une attaque qui nécessite la rendre une attention particulière. L'attaque en question évoque la possibilité de briser CDHP et BDHP en utilisant l'outil couplage de Tate (ou ses variantes), bien entendu, faire cela demande la satisfaction de quelques conditions.
L'attaque présente une menace sur l'IBE en particulier et sur la cryptographie en général. Pour lui résister, il est recommandé de travailler avec des degrés de plongement k premier ; mais cela ne peut nous servir que dans le cas des couplages symétriques, utiliser le point du premier argument d'un ordre premier et cela peut se faire en calculant par Magma (par exemple) : Factorisation $(p^k - 1)$ et choisir après r le paramètre le plus grand dans la facto-

risation, ensuite choisir arbitrairement un point P de la courbe E prenne en jeu et calculer après(|E|)/rP, la troisième condition qui a eu une grande influence sur la réussite de l'attaque et où on a jamais fait attention dans la littérature concerne le cardinal de la courbe adaptée au calcul du couplage. Une courbe de cardinal composé n'est plus souhaitable. Cette condition a chassé pas mal de courbes (adaptées au calcul du couplage) dédiée par la littérature pour la sécurité des couplages, le tableau 13 cité dans l'annexe, classe ces courbes et montre celles qui sont utiles càd qui peuvent résister à l'attaque et celles qui ne vont plus. Une courbe composée de cardinal hr où h est aussi composé de 4h', peut bien servir l'attaque puisqu'il suffit de prendre les deux points P-Q d'ordre 2r et Q d'ordre 2h'. Les courbes de Barreto-Neihrig et celles de Freeman sont les plus souhaitées pour se défendre contre cette attaque. Alors que les courbes de Brezing Weng et celles de Cocks Pinch qui sont implorées pour réussir Pairing Inversion puisqu'elles fonctionnent avec des degrés de plongement arbitraires, ne permettent de défendre que la condition 1, elles restent susceptibles devant les conditions 2 et 3. Les courbes de Barreto-Neihrig sont les éminentes dans tous les tests qu'on a faits dans le chapitre 5 et dont nous avons tenu en compte : la complexité, le format, le domaine d'utilisation et la méthode de construction des courbes.

Cela d'une part, d'autre part, comme nous visons de mettre l'IBE devant le réel (en fait, il y a déjà des sociétés qui commencent de servir leurs clients en utilisant l'IBE comme Voltage), la chose étonnante lors de notre implémentation du quatrième schéma de l'IBE est qu'il est difficile de programmer un couplage en conservant la propriété bilinéarité. Normalement, l'implémentation des couplages s'est avérée difficile par le corps cryptographes et personne ne la réussi jusqu'à aujourd'hui ; tous les efforts se concentrent sur leurs partie arithmétiques. Le visage de de notre code(s) (voir l'annexe-Listing 4) d'implémentation des couplages est à peu prés proche de celui de Dominguez Perez dans sa thèse 98 qui est dirigée par M.Scott, aussi le code de Dominguez Perez n'assure pas la bilinéarité des couplages !. On a contourné le problème en faisant des rectifications dans le schéma (dans l'attente d'une implémentation efficace).
Le chapitre 2 expose des schémas qui n'utilise pas le couplage, tandis que le chapitre 3 est réservé à ceux qui l'utilise.
Dans un pas ou nous mirons chercher le bon cryptosystème, on a prouvé par des résultats concrets l'idée de callas qui est à propos du RSA avec l'IBE sera le plus dominant. Dans la partie des schémas sous les couplages, notre nouveau IBE schéma est bien classé devant les schémas existants et ceci d'après les conclusions retirées des chapitres 3 et 5 (par des applications numériques).

Avant de terminer nous posons le monde cryptographes devant trois portes ouvertes et essentielles (la durée de cette thèse ne permet pas de chercher dans ces sens) extraites du corps de cette thèse :

- Chercher les courbes qui vérifient les quatre conditions (les trois premières sont donnés par la littérature, alors que la quatrième est ajoutée pour résister à l'attaque de la section 5.4) :

$$\begin{cases} r|p+t-1, & \\ r|p^k-1, & \text{avec r et p sont premiers} \\ Dy^2=4p-t^2, & \text{pour un entier donné y} \\ |E|=p+t-1, & \text{doit être premier.} \end{cases}$$

Il est possible de commencer avec les courbes qui sont dans la zone rachetée (voir le tableau 13) comme celles qui fonctionnent avec des degrés de plongement arbitraires.

- Trouver des méthodes qui peuvent sauver le schéma de Sakai-Kasahara et d'autres (sous l'approche Inversion) de la faiblesse expliquée dans le chapitre 5. Nous invitons la littérature de trouver des solutions à cette faiblesse comme elle l'a fait avec celle de Boneh et Franklin (une solution qu'on propose pour la faiblesse du schéma de Boneh Franklin est de prendre H(ID) d'être non un random oracle, on augmente le $H(ID)^3+aH(ID)+b$ jusqu'à ce qu'il soit un carré et en gardant le fait que le H(ID) caractérise bien le ID associé, comme cela les H(ID) ne peuvent pas se confondre ; mais dans ce cas il faut réviser la preuve de sécurité de ce schéma).

- Le point le plus important et sans lequel la cryptographie basée sur les couplages comme l'IBE n'aura pas de sens, est de trouver une méthode qui permet de bien implémenter les couplages en gardant leurs bilinéarités. Cela peut se faire en pensant peut être à une amélioration (une autre face) de l'algorithme de Miller, nous conseillons de faire un planning mathématique qui permet d'extraire des formules qui peuvent conduire à un couplage bilinéaire.

  Mais, cela ne veut pas dire qu'on ne peut pas réaliser l'IBE sans la propriété bilinéarité, puisqu'on peut prémunir le problème en utilisant quelques astuces, pourtant on dévie un petit peu le cadre de l'IBE. Par exemple avec le schéma de Boneh et Franklin :

  Pour le message chiffré C = $< rP_1, \sigma \bigoplus H_2(e(sP_1, H(ID))^r), m \bigoplus H_4(\sigma) >$, au lieu de rendre au récepteur la clé privée $d_{ID} = sH(ID)$, il suffit de le rendre srP_1 après avoir bien authentifié le récepteur qui mérite d_{ID}. Alors pour déchiffrer le message, le récepteur calcule $\sigma \bigoplus H_2(e(rsP_1, H(ID)))$ = σ au lieu de $\sigma \bigoplus H_2(e(rP_1, d_{ID}))$, ainsi, il sera facile

d'extraire les autre paramètres. La même chose sera faite avec les autres schémas (voir le chapitre 1) ; mais on remarque que le calcul de d_{ID} n'a pas d'importance ce qui demande de changer la preuve de sécurité du schéma (même chose pour les autres) de telle manière qu'on ne peut pas attaquer s (clé maîtresse).

- Dans le cadre calcul des couplages, nous recommandons aux intéressés de penser à un algorithme de Miller sous forme itérée puisqu'il peut bien réduire l'arithmétique des couplages, il peut permettre d'utiliser, facilement, ou plutôt directement les méthodes de Karatsuba, Toom Cook ainsi que d'autres.

Cela dit, cette thèse qui a comme sujet la technologie IBE, a mis l'accent sur les grands volets suivants :

- Dévoiler la plupart des axes de cette technologie.
- Suggérer des innovations importantes relatives à ce sujet
- Qualifier l'implémentation de cette technologie
- Traiter quelques unes de ses faiblesses

En fait, le sujet de notre thèse qui est le chiffrement basé sur l'identité est bien classé dans le domaine de la cryptographie où il occupe une classe importante. Il demeure utile pour différentes applications dans la vie courante :

- Flexible (sa dérivée qui est les Attributes) avec les Socials network comme Facebook et Twiters, d'après le fait qu'il utilise les clés publiques sous forme d'identité. Il peut être aussi utilisé pour sécuriser des Email.
- Utiliser au font des Smart Cards et Smart téléphoniques.
- Bien compatible pour les réseaux : Grid [128], IPv4 et IPv6..
- Sécurisations des pages Web (HTTPS) et sécuriser leurs services (Access control [147], XML) aussi.
- Communications sécurisées : intérieures et extérieures entre certains établissements publics comme les Hôpitaux, Universités, Gouvernement et Banques. Il est aussi utile pour les sociétés.
- ...

Nous renvoyons à [148] pour une culture spacieuse de tous ces services.

Annexes

Annexe A

Arithmétique performante des courbes de Barreto-Naehrig [17]

Comme nous l'avons signalé dans le chapitre 5, les courbes de Barreto-Naehring sont les échéantes d'être considérées dans l'implémentation des IBE. Dans cette annexe nous donnons des arithmétiques plus performantes pour les courbes de Barreto-Naehrig.
Ces courbes sont basées sur la donnée de :

$$p(x) = 36x^4 + 36x^3 + 24x^2 + 6x + 1$$

Et le degré de plongement k=12.
Le développement du corps $F_{p^{12}}$ (corps de base des courbes de Barreto Naehring) peut se faire en utilisant les réductions :

$$F_p \rightharpoonup F_{p^2} \rightharpoonup F_{p^{12}}$$

Nous remarquons aisément que p(x) = 1 mod 6. Ainsi nous avons -1 n'est pas un carré dans F_p, sinon, on aurait : -1=y^2 pour y un élément de F_p.
Alors

$$y^{2(\frac{p-1}{2})} = y^{p-1} = 1 \mod p \Leftrightarrow (-1)^{\frac{p-1}{2}} = 1 \mod p \Leftrightarrow (-1)^{3k} = 1 \mod p$$

Donc si k est impair on tombe dans l'absurde -2=h_1 (mod p) c.à.d -2=$h_1(1 + h_2 6)$.
Si nous prenons les p qui vérifient p=1+6k pour un k impair, on a : $x^2 + 1$ est irréductible dans F_p
Alors :

$$F_{p^2} = \{a_0 + a_1\alpha \quad \text{où} \quad (a_0, a_1) \in F_p \times F_p \quad \text{et} \quad \alpha \quad \text{est la racine de} \quad x^2 + 1 \}$$

Pour $F_{p^{12}}$ il suffit de voir que $x^6 - \beta$ est un polynôme irréductible dans F_{p^2}, pour cela, il suffit que β ne soit pas une racine sixème. Il faut chercher donc ce β.

Propriété : *Pour p un nombre premier et k un entier positif. Si p = 1 mod(k), alors un élément α pris au hasard dans F_p a une chance sur k d'être une racine k-ième dans F_p.*

On a alors une chance sur 6 que ce β existe.
Tout d'abord α ne présente pas une racine sixième d'un élément de F_p, car si cela se vérifie, on aurait :

$$\begin{aligned}\alpha &= (a_0 + \alpha a_1)^6 = ((a_0 + \alpha a_1)^2)^3 \\ &= (a_1^2\alpha^2 + 2a_0a_1\alpha + a_0^2)^3 \\ &= (a_0^2 - a_1^2)^3 + (2a_0a_1\alpha)^3 + 3(a_0^2 - a_1^2)^2 2a_0a_1\alpha \\ &\quad +3(a_0^2 - a_1^2)(2a_0a_1\alpha)^2 \\ &= (a_0^2 - a_1^2)^3 - 3(a_0^2 - a_1^2)(2a_0a_1\alpha)^2 \\ &\quad +\alpha(3(a_0^2 - a_1^2)^2 2a_0a_1 - (2a_0a_1))\end{aligned}$$

C'est un système incompatible. Par exemple, on a : $a_0 = a_1$ et $a_0a_1 = \frac{-1}{2}$
Aussi tout élément de F_{p^2} ne peut pas être une racine sixième. En effet, si cela se vérifie, on a :

$$u = u_1^6 \Longrightarrow {u_1}^{p-1} = u^{\frac{p-1}{6}} = 1 \ \ mod\, p \Longleftrightarrow u^k = 1 \ \ mod\, p$$

Alors k = $k_1(p-1)$ (il suffit d'utiliser le petit théorème de Fermat).
De p=1+6k on extrait :

$$p = 1 + 6(k_1(p-1)) \Longrightarrow p(1 - 6k_1) = 1 - 6k_1 \Longrightarrow p = 1$$

Ce qui est absurde pour les attentes cryptographiques. Alors l'élément β cherché est bien αu où u est un élément de F_p.
Le corps $F_{p^{12}}$ s'écrit alors :

$$F_{p^{12}}=\{b_0 + b_1\gamma + b_2\gamma^2 + b_3\gamma^3 + b_4\gamma^4 + b_5\gamma^5, \quad \text{où } b_0, b_1, b_2, b_3, b_4, b_5 \text{ sont dans } Fp^2 \text{ tel que } \gamma \text{ est la racine de } x^6 - \beta \ \}$$

On a :

$$(b_0 + b_1\gamma + b_2\gamma^2 + b_3\gamma^3 + b_4\gamma^4 + b_5\gamma^5)^p = b_0^p + b_1^p\gamma^p + b_2^p\gamma^{2p} + b_3^p\gamma^{3p} + b_4^p\gamma^{4p} + b_5^p\gamma^{5p}$$

Alors :

$$
\begin{aligned}
(b_0 + b_1\gamma + b_2\gamma^2 + b_3\gamma^3 + b_4\gamma^4 + b_5\gamma^5)^p &= a_{00} + a_{01}\alpha^p + (a_{10} + a_{11}\alpha^p)\gamma^p + (a_{20} + a_{21}\alpha^p)\gamma^{2p} \\
&\quad + (a_{30} + a_{31}\alpha^p)\gamma^{3p} + (a_{40} + a_{41}\alpha^p)\gamma^{4p} \\
&\quad + (a_{50} + a_{51}\alpha^p)\gamma^{5p} \\
&= a_{00} + a_{01}\alpha(-1)^k + (a_{10} + a_{11}\alpha(-1)^k)\gamma\beta^k \\
&\quad + (a_{20} + a_{21}\gamma)\gamma^2\beta^{2k} + (a_{30} + a_{31}\alpha(-1)^k)\gamma^3\beta^{3k} \\
&\quad + (a_{40} + a_{41}\alpha(-1)^k)\gamma^4\beta^{4k} + (a_{50} + a_{51}\alpha(-1)^k)\gamma^5\beta^{5k} \\
&= a_{00} + a_{01}\alpha(-1)^k + (a_{10} + a_{11}\alpha(-1)^k)\gamma(u\alpha)^k \\
&\quad + (a_{20} + a_{21}\gamma)\gamma^2(u\alpha)^{2k} + (a_{30} + a_{31}\alpha(-1)^k)\gamma^3(u\alpha)^{3k} \\
&\quad + (a_{40} + a_{41}\alpha(-1)^k)\gamma^4(u\alpha)^{4k} \\
&\quad + (a_{50} + a_{51}\alpha(-1)^k)\gamma^5(u\alpha)^{5k}
\end{aligned}
$$

Tout dépend alors de k (pair ou impair) on peut obtenir des calculs encore plus minimisant. Ces calculs sont utiles pour le cacul d'une puissance sous forme Frobeinus.

Pour l'inverse il suffit de remarquer que :

$$(b_0 + b_1\gamma + b_2\gamma^2 + b_3\gamma^3 + b_4\gamma^4 + b_5\gamma^5)^{p-1} = (b_0 + b_1\gamma + b_2\gamma^2 + b_3\gamma^3 + b_4\gamma^4 + b_5\gamma^5)^{6k}$$

Alors :

$$(b_0 + b_1\gamma + b_2\gamma^2 + b_3\gamma^3 + b_4\gamma^4 + b_5\gamma^5)^{-1} = \frac{(b_0+b_1\gamma+b_2\gamma^2+b_3\gamma^3+b_4\gamma^4+b_5\gamma^5)^{6k}}{(b_0+b_1\gamma+b_2\gamma^2+b_3\gamma^3+b_4\gamma^4+b_5\gamma^5)^p}$$

Il suffira alors de faire la division euclidienne après avoir développé le numérateur en bénéficiant de γ^6 = $u\alpha$ et de la relation de Horner :

$$b_0 + \gamma(b_1 + \gamma(b_2 + \gamma(b_3 + \gamma(b_4 + \gamma(b_5)))))$$

Annexe B

Algorithme de Barreto-Naehring

Les courbes de Barreto-Naehring peuvent être générées simplement en déroulant l'algorithme suivant :

Listing 1 : Code en Magma de l'algorithme de Barreto-Naehring [17]

```
x :=Random;
    while true do
       t := 6*x^2 + 1;
       p := 36*x^4 - 36*x^3 + 24*x^2 - 6*x + 1; //P(-x)
       n := p + 1 - t;
        if IsPrime(p) and IsPrime(n) then
          break;
        end if;
       p := 36*x^4 + 36*x^3 + 24*x^2 + 6*x + 1; //P(x)
       n := p + 1 - t;
        if IsPrime(p) and IsPrime(n) then
          break;
        end if;
       x := x + 1;
    end while;
Fp := FiniteField(p);
b := Fp!0;
    repeat
      repeat
        b := b + 1;
      until IsSquare(b + 1);
        y := Fp!Root(b + 1, 2);
        E := EllipticCurve([Fp|0,b]);
        G := E![1,y];
    until IsIdentity(n*G);
"Resultat : Fp = ", Fp, " E = ", E, " b = ", b, " G = ", G;
```

Pour n=64, on obtient :

E = courbe elliptique définie par : $y^2 = x^3 + 3$ sur le corps :

GF(41685152125433846774136946582933352672728954774085062662281567928883759704049 63)

G = (1 : 41685152125433846774136946582933352672728954774085062662281567928883759704049 61 : 1)

t = 204169420152563118299305542775369251293 5

n= 41685152125433846774136946582933352672708537832069806350451637374606222778920 29

Pour n=170, on obtient :

E = courbe elliptique définie par : $y^2 = x^3$ + 10 sur le corps :

GF(18059243436407227159564555719727004238110755243553963901993504009969780627713888 69647354009734010084771408928254513487473097100824653744555570834464532284228445997 32220305562008626071049442945611419868083593)

G = (1 :12748468606020879660827463162356238393547576317912935966868901687307200838170 85685730271830366825181997330723712638826375485053828657920270880372409643040029394805354912525008508620752172983886342051631142 8 : 1)

t = 13438468453066825263344653683410665668729872983443886909367456964440760533046698714519311286202197323105

n= 18059243436407227159564555719727004238110755243553963901993504009969780627713888696473540097340100847712745435699828192204636542878196378988835357346978453593523227525586480147557937233492363432521767076048 9

Annexe C

Méthode de Multiplication Complexe

Nous avons parlé dans le chapitre 5 de la construction des courbes adaptées au calcul des couplages qui est basée sur la méthode de Multiplication Complexe (voir la partie 5.5.1). Nous détaillons dans le suivant la stratégie de cette méthode dans le cas des courbes MNT où le degré de plongement k=3 et D=19.

Génération de la courbe MNT dans le cas où k=3 et D=19

Les courbes MNT dans le cas où k=3 se caractérisent par : t = $1 \pm 6s$ et q = $12s^2$ - 1.

Après le remplacement dans l'équation $4q^2 - t^2 = Dy^2$, on obtient :

$$4q^2 - t^2 = Dy^2 \Longleftrightarrow x^2 - 3Dy^2 = 24 \ \ où \ \ x = 6s \pm 3 \tag{73}$$

Pour chercher une solution à l'équation 73, on cherche tout d'abord une solution particulière (solution minimale) à l'équation $x^2 - 3Dy^2 = 1$ et puis une solution générale à : $x^2 - 3Dy^2 = 24$.

Une solution minimale de $x^2 - 3Dy^2 = 1$ vérifie :

$$|\frac{x_0}{y_0} - \sqrt{3D}| < \frac{1}{2y_0^2} \tag{74}$$

Elle peut être obtenue en détectant, dans le développement en fractions continues de $\sqrt{3D}$, la première réduite $\frac{x_0}{y_0}$ pour un x_0 et y_0 qui vérifient ${x_0}^2 - 3D{y_0}^2 = 1$

L'algorithme suivant permet de calculer une solution minimale (x,y) de l'équation de Pell : $x^2 - ny^2 = 1$ (réduites du développement en fractions continues de $\sqrt{n}$).

Listing 2 : Code en Magma de l'algorithme qui calcule une solution minimale de $x^2 - ny^2 = 1$

```
n :=Random;
a0 := Truncate(Sqrt(n)); a :=a0; b :=0;
ct :=n; c :=1;
xt :=1; x :=a0;
yt :=0; y :=1;
    repeat
      bp := a*c - b;
      cp := 2*a*b-(a^2)*c + ct;
      a := Truncate((a0+bp)/cp);
```

```
        xp := a*x + xt;
        yp := a*y + yt;
        b := bp;
        ct := c; c := cp;
        xt := x; x := xp;
        yt := y; y := yp;
    until (x*x - n*y*y) eq 1;
"Résultat : x = ", x, "et y = ", y;
```

Une fois trouvé les solutions (x_0, y_0) de l'équation de Pell particulier $x^2 - 3Dy^2 = 1$ en utilisant cet algorithme, on cherche exaustivement ensuite (x_p, y_p) les solutions de l'équation de Pell généralisé : $x^2 - 3Dy^2 = 24$. Après, on construit une famille de solutions (u[1], u[2]) qui vérifie :

$$u[1] + \sqrt{3D}u[2] = (x_p + \sqrt{3D}y_p)(x_0 + \sqrt{3D}y_0)^k$$

(il suffit d'utiliser le listing 3 pour trouver u[1]) à condition que les solutions trouvées vérifient :

$$u[1] = 6x \pm 3 \quad \text{et} \quad q = 12(\tfrac{u[1]\pm 3}{6})^2 - 1 \quad \text{soit} \quad \text{premier.}$$

Avec un $12 * (((u[1] - 3)/6)^2) - 1$ où u[1]= *1381693086544638334867260819* on génère le nombre : q = *6363585951350831465366416367529396268971941369896619 51*. Ce nombre est bien premier (vérification par IsPrime). Alors le corps fini qu'on puisse prendre est : F_q
où q = 636358595135083146536641636752939626897194136989661951.
Le polynôme de Hilbert pour D=-19, se calcule par : HilbertClassPolynomial(-19)=1 + 884736.
En utilisant *L!-884736;*, on obtient :*6363585951350831465366416367529396268971941369887772 15* comme étant racine du polynôme de Hilbert et par :

L!((6363585951350831465366416367529396268971941369887772 15)/(1728-
6363585951350831465366416367529396268971941369887772 15));

on obtient : k_1=*1498270077253852041202250394997502988514931793601809 65;* qui est le paramètre à partir du quel on peut construire $y^2 = x^3 + 3k_1c^2x + 2k_1c^3$ où c est quelconque dans F_q où q = 636358595135083146536641636752939626897194136989661951.
L'équation de la courbe recherchée pour c=2 est :

y^2=x^3 +*5252069024344561563694172004911243324235298783428476 78x* +
4881563382009138263136757217371859009323084587939095 87

Listing 3 : Code pour calculer $(x_p + \sqrt{D}y_p)(x_0 + \sqrt{D}y_0)^k \implies$ *q premier*

Dans le cas de la courbe MNT avec un k=3 et D=19

```
param1 := function(i)
 x :=x_p ; y :=y_p ; z :=x_0 ; t :=y_0 ;
  for n in [1..i-1] do
   v :=[x*z+y*t*57,x*t+y*z,i] ;
   z :=v[1] ;
   t :=v[2] ;
  end for ;
 return [z,t, i] ;
end function ;
  for i in [1..j-1] do
     u :=[x*z*param1(n)[1]+param1(n)[2]*3*D,x*z*param1(n)[2]+param1(n)[1]] ;
     IsPrime(12*(((u[1]+3)/6)^2)-1) ; %(where u[1]^2 - 3 * D * u[2]=24)
       else
        IsPrime(IsPrime(12*(((u[1]-3)/6)^2)-1) ;) ;
  end for
```

Un exemple de courbes MNT qui ont un degré de plongement k = 6 et dans le cas où D=43 (ils ont les paramètres : $q = 4s^2 + 1$ et $t = 1 \pm 2s$ qui se ramènent à la résolution de l'équation : $x^2 3Dy^2 = 8$) peut être généré par la méthode de multiplication complexe sous la forme :

$y^2 = x^3 + 1763476217229032x + 3447467182151685$ dans le corps F_q

Où q = 4592547906355601

Pour construire d'autres courbes MNT (selon le D convenable) nous conseillons de consulter le site : [1]

1. http ://crypto.stanford.edu/pbc/mnt.html

Annexe D

Classement des courbes bien adaptées

Nous présentons ci-dessous le classement des courbes bien adaptées depuis leurs apparitions et suivant la résistance devant l'attaque présentée dans 5.4. Pour un état de l'art sur ces courbes nous renvoyons à [79].

Courbes	Résistance à l'attaque	Raisons	Degré de résistance	Solution proposée
Barreto-Naehering [17]	Oui	Ordre premier	Bonne résistance	-
Cocks Pinch [59]	Il dépend	Degré de plongement arbitraire	Il dépend	Courbes avec des ordres purement premiers
MNT [139]	Il dépend	-	-	-
Freeman [80]	Oui	Ordre premier	Bonne résistance	-
Dupong-Eng-Morain [66]	Oui	Ordre premier	Bonne résistance	-
Scott-Barreto [160]	Oui/Non	Cofacteur petit	Résistance moyenne	Cofacteur h=1
Kachisa-Schaefer-Scott [114]	Non	Existance d'un cofacteur e	Mauvaise résistance	Cofacteur e grand et impair (cas de k=40)
Barreto-Lynn-Scott [16]	Non	Cofacteur petit	Mauvaise résistance	Courbes sous forme MNT
Galbraith-Mckee-Valenca [84]	Non	Cardinal composé	Mauvaise résistance	Aucunne proposition
Boneh-Rubin-Silverberg [29]	Il dépend	Cardinal muni d'une composition appropriée	Il dépend	Composé plus grand(difficile)
Brezing-Weng [43]	Il dépend	Degré de plongement arbitraire	Il dépend	Courbes avec des ordres purement premiers

TABLE 13 – Classement des courbes construitent pour servir les couplages bien adaptés.

■ : Couleur réserver aux courbes qui résistent bien à l'attaque

■ : Cette couleur est pour les courbes qui présentent un danger

■ : Elle est pour les courbes qu'on peut rectifier

Pour pouvoir obtenir un résultat s'approchant des courbes colorées en bleu, il serait préférable de chercher les courbes vérifiant les conditions suivantes (les trois premières sont données par la littérature) :

$$\begin{cases} r|p+t-1, & \\ r|p^k-1, & \text{avec r et p sont premiers} \\ Dy^2=4p-t^2, & \text{pour un entier donné y} \\ |E|=p+t-1, & \text{doit être premier.} \end{cases}$$

Annexe E

Implémentation des couplages

Dans les implémentations des IBE, nous avons fait usage au code ci-dessous pour implémentater le couplage. Nous donnons ici le visage du code dans le cas du couplage de Weil (le code est aussi valable pour le couplage de Tate).

Listing 4 : Code en Magma du couplage de Weil en utilisant les coordonnées Affines

```
L :=function(A,B,Q,a)
  if A[1] eq B[1] then
    if A[2] eq 0 then
       return Q[1]-A[1];
    end if;
     lambda :=(3*A[1]^2 + a)/ (2*A[2]);
    return Q[2]-A[2]-Q[1]*lambda+A[1]*lambda;
        else if B[1] eq A[1] then
           return Q[1]-B[1];
        end if;
           lambda :=(B[2]-A[2])/ (B[1]-A[1]);
    end if;
  return (lambda*(-Q[1]+B[1]) - B[2]+Q[2]);
end function;
Miller := function(P,Q,r,a)
  f :=1 ; g :=1 ; T :=P;
  i :=Floor(Log(2,r))-1;
  si :=Intseq(r,2);
  while i ge 0 do
    if T[2] eq 0 then
      break;
    end if;
 lambda :=(3*T[1]+a)/(2*T[2]);
  f :=f^2*L(T,T,Q,a);
  g := g^2*(Q[1]-lambda^2+2*T[1]);
  T :=[lambda^2-2*T[1], lambda*(T[1]- lambda^2+2*T[1])-T[2]];
    if si[i+1] eq 1 then
```

```
        if T[1] eq P[1] then
          break;
        end if;
      lambda :=( T[2] - P[2])/ ( T[1] - P[1]);
      f :=f*L(T,P,Q,a);
      g :=g*(Q[1]-lambda^2+T[1]+P[1]);
      T :=[lambda^2- T[1]-P[1],lambda*(T[1]- lambda^2+T[1]+P[1])-T[2]];
       end if;
      i :=i-1;
     end while;
    return f/g;
  end function;
```

En utilisant la courbe suivante de Freeman :

$y^2 = x^3+1464499331930237340780000436644371333362x+76245224369580602590094637750757166227 78$

Définie sur la le corps $F_{p^{10}}$, où p = 85077849915766157069541947893248322674 43 (132-bits) :
Aussi, en utilisant les paramètres :

```
P :=E![1200520319794122551047086306997980247063,
477575087879488930569817501869180957801, 1];

r :=1057077544124515873508312074 1;

Q :=G![8141375668928953090248682170713439618763*K.1^9 +
39034085082431875629967843894199786417 36*K.1^8 +
24274512348696512591861432770817802409 09*K.1^7 +
44546795669732562802459885470806342121 49*K.1^6 +
19470760032318510649425689813748950672 24*K.1^5 +
49724449871366996045969438636897785428 9*K.1^4 +
86223824407822109128504891749735458838 2*K.1^3 +
35366057073265314211929361108454414177 96*K.1^2 +
18866373668152156941409062083654225823 2*K.1 +
79416654761093997671236558583010494875 67,
53699054988516607203077423228844411696 7*K.1^9 +
40345481192630221688674514550858086706 4*K.1^8 +
38063861117313896116789082825620986493 22*K.1^7 +
18020764197876994276723456591119869157 25*K.1^6 +
33619590730963987519723795378475804245 18*K.1^5 +
```

436157850351731386859193064023913284987 9*K.1^4 +
410686198531267871961261847224249640657 1*K.1^3 +
388230998590192795817911570371309022320 9*K.1^2 +
180829984579948159105738298903577859220*K.1 +
607881878181351687480708785809834439571 2, 1];

On obtient, le calcul suivant du couplage de Weil :

479709511231047331609900015149666681168 7*K.1^9 +
163166250475868502503299445683851356159 7*K.1^8 +
467368705155372481903703750644919595309 4*K.1^7 +
116611075662041876197913454133189561431 0*K.1^6 +
442659945834753017408540329699798988170 7*K.1^5 +
825665836425276802283909805558213009177 9*K.1^4 +
502087323394337762066951132541234296746 1*K.1^3 +
674569082274194015104775392354468336181 0*K.1^2 +
804196572522091291127372042612565623249 3*K.1 +
503305362017852512295567790231857558926 5

Calculé dans un temps 0.540 seconds

Publications actuelles

Articles Journaux avec comité de sélection et actes :

1. Rkia Aouinatou, Mostafa Belkasmi. ***On the conventional of pulic key in Identification Based Encryption : the case of RSA***. *pp. 171-201, vol 5, num 3. International Journal of Information and Computer Security, Inderscience. DOI : 10.1504/IJICS.2013.055-837.*
2. Rkia Aouinatou, Mostafa Belkasmi, Mohamed Askali. ***A dynamic study with side channel against Identification Based Encryption***. *pp. 8-19, vol 7, num 1. International Journal of Communication Networks and Information Security (IJCNIS).*

Articles Journaux soumis :

3. Rkia Aouinatou, Mostafa Belkasmi. ***Cryptanalysis CDHP, BDHP and Tate pairing. Under certain condition the Tate pairing is less secure than Weil*** *soumis le 19/1/2015 dans Journal of Cryptology.*
4. Rkia Aouinatou, Mostafa Belkasmi. ***An efficient and secure Identity Based Encryption in the Model Random Oracle***. *Soumis*
5. Rkia Aouinatou, Mostafa Belkasmi. ***Toward two efficient schemes IBE and HIBE in the model selective ID***. *En soumission.*

Conférences Internationales (avec comité de sélection et actes) :

6. Rkia Aouinatou, Mostafa Belkasmi. ***Efficient Anonymity for Coks Scheme***. *Dans ISIVC10 (IEEE), Rabat, octobre, 2010.*
7. Rkia Aouinatou, Mostafa Belkasmi. Kamelia Kadri. ***IBE and the Attack DPA Between the Existence and the Reality***. *Dans ICMCS11 (IEEE), Ouarzazate, Avril, 2011.*

Reports sous IACR eprint archive ou CORR :

8. Rkia Aouinatou, Mostafa Belkasmi. ***Identification Based Encryption with RSA- OAEP. Using SEM and Without***. *IACR eprint archive, report 2011/684.*

9. Rkia Aouinatou, Mostafa Belkasmi. ***In the point of view security. An efficient scheme with random oracle***. *IACR eprint archive, report 2012/237.*
10. Rkia Aouinatou, Mostafa Belkasmi. ***Cryptanalysis CDHP, BDHP and Tate pairing. Under certain condition the Tate pairing is less secure than Weil***. *IACR eprint archive, report 2012/277.*
11. Rkia Aouinatou, Mostafa Belkasmi. ***An efficient classification provid with an aprovement of BB1 to an efficient commutative blinding scheme***. *Report : CoRR abs/1208-1217/2012/.*

Conférence Nationale (avec comité de sélection) :

12. Rkia Aouinatou, Driss Aboutajdinne, Mostafa Belkasmi. ***Une méthode contre Key Escrow en IBE***. *Dans JDTIC10, Fès, 2010.*

Workshops Internationaux (avec comité de sélection) :

13. Rkia Aouinatou, Mostafa Belkasmi. ***A competitive in Identification Based Encryption Which Cryptosystems that can succed***. *Dans WOTIC11, Casa, Octobre, 2011.*
14. Rkia Aouinatou, Mostafa Belkasmi. ***Un modèle plus efficace du Commutative Blinding : Une amélioration de BB1***. *Dans WOTIC11, Casa, Octobre , 2011.*

Bibliographie

[1] Agrawal, M., Kayal, N,. Saxena, N. *PRIMES is in P*. Annals of Mathematics. Vol. 160. Num. 2. pp. 781-793. 2004.

[2] Al-Riyami, S.S., Paterson, K. G. *Certificateless public key cryptography*. Dans ASIACRYPT. Vol. 2894 of Lecture Notes in Computer Science. Springer-Verlag. pp. 452-473. 2003.

[3] Aouinatou, R., Belkasmi, M. *Efficient Anonymity for Coks Scheme*. Dans ISIVC10 (IEEE), Rabat, octobre, 2010.

[4] Aouinatou, R., Belkasmi, M. *On the conventional of pulic key in Identification Based Encryption : the case of RSA*. International Journal of Information and Computer Security. Vol. 5. Num. 3. pp. 171-201. 2013. DOI : 10.1504/IJICS.2013.055-837. Aussi available sur IACR eprint archive, report 2011/684.

[5] Aouinatou, R., Belkasmi, M. *Toward two efficient schemes IBE and HIBE in the model selective ID*. En soumission. Available sur CoRR abs/1208-1217/2012/.

[6] Aouinatou, R., Belkasmi, M. *An efficient and secure Identity Based Encryption in the Model Random Oracle*. Soumis . Available sur IACR eprint archive, report 2012/237.

[7] Aouinatou, R., Aboutajdinne, D., Belkasmi, M. *Une méthode contre Key Escrow en IBE*. Dans JDTIC10, Fès, 2010.

[8] Aouinatou, R., Belkasmi, M., Askali, M. *A dynamic study with side channel against Identification Based Encryption*. International Journal of Communication Networks and Information Security (IJCNIS). Vol. 7. Num. 1. pp. 8-19. 2015.

[9] Aouinatou, R., Belkasmi, M., Aboutajddine, D. *Cryptanalysis CDHP, BDHP and Tate pairing. Under certain condition the Tate pairing is less secure than Weil*. Soumis le 19/1/2015 dans Journal of Cryptology. Springer (IACR). Aussi available sur : IACR eprint archive, report 2012/277.

[10] Ateniese, G., Gasti, P. *Universally Anonymous IBE based on the Quadratic Residuosity Assumption*. CT-RSA '09 Proceedings of the The Cryptographers' Track at the RSA Conference 2009 on Topics in Cryptology. pp. 32-47. 2009.

[11] Atkin, A. O. L., Morain, F. *Elliptic curves and primality proving*. Math. Comp. Vol. 61. Num. 203. pp. 29-68. 1993.

[12] Attrapadung, N., Chevallier-Mames, B., Furukawa, J., Gomi, T., Hanaoka, G., Imai, H., Zhang, R. *Efficient identity based encryption with tight security reduction*. Cryptology ePrint Archive, Report 2005/320, 2005.

[13] Bajard, J.C., El Mrabet, N. *Pairing in cryptography : an arithmetic point of view*. Advanced Signal Processing Algorithms, Architectures, and Implementations XVII, part of the SPIE Optics & Photonics 2007 Symposium (Proceedings of SPIE). Vol. 6697. pp. 66970O.1-66970O.11. August 2007.

[14] Barthe, G., Olmedo, F., and Zanella Bguelin, S. *Verifiable security of boneh-franklin identity-based encryption*. 5th International Conference on Provable Security ProvSec 11. Lecture Notes in Computer Science. Springer. 2011.

[15] Barreto, P.S.L.M., Lynn, B., Scott, M. *On the selection of pairing-friendly groups*. M. Matsui and R. Zuccherato, Editors, Selected Areas in Cryptography - SAC 2003. Vol. 3006 of Lecture Notes in Computer Science. Springer-Verlag. pp. 17-25. 2004.

[16] Barreto, P.S.L.M., Lynn, B., Scott, M. *Constructing elliptic curves with prescribed embedding degrees*. Dans Security in Communication Networks SCN. Third International Conference, SCN02. Vol. 2576 of Lecture Notes in Computer Science. Springer Verglas. pp. 257-267. 2002 (Amalfi, Italy, September 1113, 2002). DOI. 10.1007/3-540-36413-7_19.

[17] Barreto, P.S.L.M., Naehrig, M. *Pairing-friendly elliptic curves of prime order*. Selected Areas in Cryptography SAC05. Vol. 3897 of Lecture Notes in Computer Science. (Springer, Berlin, 2006). pp. 319-331. 2005.

[18] Boldyreva, A., Goyal, V., Kumar, V. *Identity-based encryption with efficient revocation*. ACM Conference on Computer and Communications Security. pp. 417-426. 2008

[19] Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G. *Public Key Encryption with Keyword Search*. Advances in Cryptology, EUROCRYPT 04. Vol. 3027 of Lecture Notes in Computer Science. Interlaken, Switzerland. pp. 506-522, 2004.

[20] Boneh, D., Ding, X., Tsudik, G., Wong, C. M. *A method for fast revocation of public key certificates and security capabilities*. 10th USENIX Security Symposium, Washington, D. C., Aug. 2001.

[21] Boneh, D., Lynn, B., Shacham, H. *Short signatures from the Weil pairing*. Advances in Cryptology Asiacrypt 2001. Vol. 2248 of Lecture Notes in Computer Science (Springer, Berlin, 2002). pp. 514-532. Full version : Journal of Cryptology. Vol. 17. pp. 297-319. 2004.

[22] Boneh, D., Franklin, M. *Identity-based encryption from the Weil pairing*. In Advances in Cryptology Crypto 2001. Vol. 2139 of Lecture Notes in Computer Science (Springer, Berlin, 2001). pp. 213-229. Full version : SIAM J. Comput. Vol. 32. Num. 3. 586-615. 2003.

[23] Boneh, D., Boyen, X. *Efficient selective-ID secure identity based encryption without random oracles*. Dans Christian Cachin and Jan Camenisch, Editors. Advances in Cryptology - EUROCRYPT04. Vol. 3027 of Lecture Notes in Computer Science. Springer-Verlag. pp. 223-238. 2004.

[24] Boneh, D., Boyen, X. *Secure identity based encryption without random oracles*. In Crypto 04. Vol. 3152 of Lecture Notes in Computer Science. Springer. pp. 443-459. 2004.

[25] Boyen, X. *The BB1 identity-based cryptosystem : A standard for encryption and key encapsulation*. IEEE 1363.3, aug 2006. Available sur :http ://grouper.ieee.org/groups/1363/.

[26] Boyen, X. *A tapestry of identity-based encryption : Practical frameworks compared*. International Journal of Applied Cryptography. Vol. 1. Num. 1. pp. 3-21, 2008.

[27] Boneh, D., Katz, J. *Improved efficiency for CCA-secure cryptosystems built using identity based encryption*. Topics in Cryptology-CT-RSA 05, San Francisco. CA. USA. February 14-18, 2005. Proceedings. Vol. 3376 of Lecture Notes in Computer Science. Springer. pp. 87-103. 2005.

[28] Boneh, D., Ding, X., Tsudik, G. *Identity based encryption using mediated RSA**. 3rd Workshop on Information Security Application, Jeju Island, Korea, Aug. KIISC. 2002.

[29] Boneh, D., Rubin, K., Silverberg, A. *Finding composite order ordinary elliptic curves using the Cocks-Pinch method*. Journal of Number Theory. Vol. 131. Num. 5 (May 2011). pp. 832-841. 2011. Aussi available sur : IACR Cryptology ePrint Archive 2009/533.

[30] Boneh, D., Boyen, X., Goh, E.J. *Hierarchical identity based encryption with constant size ciphertext*. Dans Ronald Cramer, Editor, Advances in Cryptology - EUROCRYPT05. Vol. 3494 of Lecture Notes in Computer Science. Springer-Verlag. pp. 440-456. 2005.

[31] Boxall, J., Enge, A. *Some security aspects of pairing-based cryptography*. ANR Project PACE. Pairings and Advances in Cryptology for. E-cash. October 3, 2009.

[32] Boyen, X., Martin, L. *Identity-Based Cryptography Standard (IBCS) #1 : Supersingular Curve Implementations of the BF and BB1 Cryptosystems*. Available sur : http ://2rfc.net/5091.

[33] Boldyreva, A., Fischlin, M. *On the Security of OAEP*. Advances in Cryptology ASIACRYPT06. Vol. 4284 of Lecture Notes in Computer Science. Springer-Verlag. pp. 210-225. 2006.

[34] Bellare, M., Rogaway, P. *Optimal Asymmetric Encryption How to encrypt with RSA*. Dans Eurocrypt 94 Proceedings. Springer-Verlag. Lecture Notes in Computer Sciences 950. pp. 92-111. 1995

[35] Bretonnière, L. *Fonctions modulaires et extensions de corps quadratiques imaginaires*. Laboratoire de Mathématiques Nicolas Oresme, CNRS URM 6139 Caen Cedex, France. Mémoire de Master 2 Mathématique Fondamentales.

[36] Bellare, M., Ristenpart, T. *Simulation without the Artificial Abort : Simplified Proof and Improved Concrete Security for Waters IBE Scheme*. Cryptology ePrint Archive : Report 2009/084.

[37] Bellare, M., Desai, A., Pointcheval, D., Rogaway, P. *Relations Among Notions of Security for Public-Key Encryption Schemes*. Dans H. Krawczyk, Editeur. Advances in Cryptology-Crypto 98, Vol. 1462 of Lecture Notes in Computer Science. Springer Verlag. pp. 26-45. 1998.

[38] Bellare, M., Rogaway, P. *Random Oracles are Practical : A Paradigm for Designing Efficient Protocols*. Proc. First ACM Conf. Computer and Comm. Security (CCS 93). pp. 62-73. 1993.

[39] Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D. *Key-Privacy in Public-Key Encryption*. Dans Advances in Cryptology ASIACRYPT. Lecture Notes in Computer Science. Springer Verlag. 2001

[40] Bellare, M. Rogaway, P. *Optimal asymmetric encryption*. Dans : EUROCRYPTO94. Vol. 839 of Lecture Notes in Computer Science. Springer Verlag. pp. 92-111. Heidelberg. 1994.

[41] Brier, E., Coron, J.S., Icart, T., Madore, D., Randriam, H., Tibouchi, M. : *Efficient indifferentiable hashing into ordinary elliptic curves*. Dans Advances in Cryptology CRYPTO10. Vol. 6223 of Lecture Notes in Computer Science. Springer Verlag. pp. 237-254. 2010.

[42] B. Buchberger. *Gröbner bases : An algorithmic method in polynomial ideal theory*. Dans N. Bose, Editor. Multidimensional systems theory, Progress, directions and open problems, Math. Appl. Vol. 16. pp. 184-232. D. Reidel Publ. Co. 1985.

[43] Brezing, F., Weng, A. *Elliptic curves suitable for pairing based cryptography*. Des Codes and Cryptography. Vol. 37. pp. 133-141. 2005.

[44] Blake, I., Seroussi, G., Smart, N. editors. *Advances in Elliptic Curve Cryptography*. London Mathematical Society 317, Cambridge University Press, 2005. CHAPITRE IX. Galbraith.

[45] Canetti, R., Halevi, S., Katz, J. *Chosen-ciphertext security from identity-based encryption*. Dans Advances in Cryptology EUROCRYPT. Vol. 3027 of Lecture Notes in Computer Science. Springer Verlag. pp. 207-222. 2004.

[46] Callas, J. *Identity-Based Encryption with Conventional Public-Key Infrastructure*. Dans 4th Annual PKI R&D Workshop, number 7224 in Interagency Reports, pp. 102-115. NIST. 2005.

[47] Charlemagne, M. *Pairings in cryptology : efficiency, security, and applications*. Dublin City University, Thèse. 2010.

[48] Cha, J., Cheon, J.H. *An Identity-Based Signature from Gap Diffie-Hellman Groups*. Dans Public Key Cryptography-PKC03. Vol. 2567 of Lecture Notes in Computer Science. Springer Verlag. pp. 18-30. 2003.

[49] Chen, L., Cheng, Z. *Security Proof of Sakai-Kasaharas Identity-Based Encryption Scheme*. Dans Proceedings of Cryptography and Coding 2005.

[50] Chevallier-Mames, B,. Joye, M. *Chosen-Ciphertext Secure RSA-type Cryptosystems*. Dans J. Pieprzyk and F. Zhang, Editors. Provable Security (ProvSec 2009). Vol 5848 of Lecture Notes in Computer Science. Springer Verlag. pp. 32-46. 2009.

[51] Chevallier-Mames, B. *Cryptographie à clé publique : Constructions et preuves de sécurité*. École Normale Supérieure, 2006. Thèse, available sur : http ://bcm.crypto.free.fr/pdf/PhD.

[52] Cheon, J. *Security analysis of the strong Diffie-Hellman problem*. Dans Serge Vaudenay, Editor. EUROCRYPT 2006. Vol 4004 of Lecture Notes in Computer Science. Springer Verlag. pp. 1-11. Berlin, Germany, May / June 2006.

[53] Chen, L., Cheng, Z., Malone-Lee, J., Smart, N. *An efficient id-kem based on the sakai-kasahara key construction*. Cryptology ePrint Archive, Report 2005/224. 2005. http ://eprint.iacr.org/.

[54] Chen, L., Kudla, C. *Identity based authenticated key agreement from pairings*. Cryptology ePrint Archive, Report 2002/184. 2002. http ://eprint.iacr.org/ 2002/184.

[55] Cocks, C. *An identity based encryption scheme based on quadratic residues*. Proceedings of IMA 2001. Vol. 2260 of Lecture Notes in Computer Science. Springer Verlag. pp. 360-363. 2001.

[56] Cramer, R., Shoup, V. *Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack*. SIAM Journal on Computing. Vol. 33. Num. 1. pp. 167-226. 2003.

[57] Coron, J. S. *Resistance Against Differential Power Analysis for Elliptic Curve Cryptosystems*. Dans C .K. Koc and C. Paar, Editors. Cryptographic Hardware and Embedded Systems. Vol. 1717 of Lecture Notes in Computer Science. Springer Verlag. pp. 292-302. 1999.

[58] Costello, C., Boyd, C., González Nieto. J. M., Koon-Ho Wong, K. *Avoiding Full Extension Field Arithmetic in Pairing Computations*. Progress in Cryptology-AFRICACRYPT 2010. Vol. 6055 of Lecture Notes in Computer Science. Springer Verlag. pp. 203-224. 2010.

[59] Cocks, C., Pinch, R.G.E. *Identity-based cryptosystems based on the Weil pairing*. Unpublished manuscript, 2001.

[60] Dai, W. *Crypto++ library*. http ://www.cryptopp.com.

[61] Desmedt, Y., Quisquater, J. J. *Public-key systems based on the difficulty of tampering*. Dans Crypto 86. Vol. 263 of Lecture Notes in Computer Science. Springer Verlag. pp. 111-117. 1986.

[62] De Canniere, C., Mendel, F., Rechberger, C. *Collisions for 70-Step SHA-1 : On the Full Cost of Collision Search*. Dans Carlisle M. Adams, Ali Miri, and Michael J. Wiener, Editors. Selected Areas in Cryptography. Vol. 4876 of Lecture Notes in Computer Science. Springer Verlag. pp. 56-73. 2007.

[63] Diffie, W., Hellman, M. E. *New directions in cryptography*. IEEE Transactions on Information Theory. Vol. 22. Num. 6. pp. 644-654. 1976.

[64] Ding, X., Tsudik, G. *Simple Identity-Based Cryptography with Mediated RSA**. Dans Proceedings of CT-RSA'03. Vol. 2612 of Lecture Notes in Computer Science. Springer Verlag. 2003.

[65] Dominguez Perez, L. J. *Developing an automatic generation tool for cryptographic pairing functions*. Faculty of Engineering and Computing School of Computing. Thèse. 2011. Available sur : http ://doras. dcu.ie/16002/1/thesis.pdf

[66] Dupont, R., Enge, A., Morain, F. Building Curves with Arbitrary Small MOV Degree over Finite Prime Fields. Journal of Cryptology. Vol. 18. Num. 2. pp. 79-89. 2005. IACR Cryptology ePrint Archive 2002 : 94 (2002).

[67] Duquesne, S. *RNS arithmetic in Fpk and application to fast pairing computation*. Journal of Mathematical Cryptology. Vol. 5. Num. 1. pp. 51-88. 2011.

[68] ECRYPT II (European Network of Excellence in Cryptology II) ECRYPT2. *YearlyReport on Algorithms and Keysizes (2011-2012)*. Available sur le site :http ://ec.europa.eu/informatonsociety/apps/projects/logos/6/216676/080/deliverables/001-DSPA20.pdf

[69] El Mrabet, N. *What about Vulnerability to a Fault Attack of the Miller's Algorithm During an Identity Based Protocol ?*. Third International Conference andWorkshops, Advances in Information Security and Assurance. Vol. 5576 of Lecture Notes in Computer Science. Springer Verlag. pp. 122-134, Seoul, Korea. 2009.

[70] El Mrabet, N. *Arithmétique des couplages, performance et résistance aux attaques par canaux cachés*. Université Montpellie II - Sciences et Techniques du Languedoc. Thèse. December 2009

[71] El Mrabet, N., Natale. G. Di., Flottes, M. L. *A practical diferential power analysis attack against the miller algorithm*. Dans PRIME 2009 - 5th Conference on Ph.D. Research in Microelectronics and Electronics, Circuits and Systems Magazine. IEEE Xplore. 2009.

[72] Ellison, C., Schneier, B. *Ten Risks of PKI : What You're Not Being Told about Public Key Infrastructure*. Journal of Computer Security. Vol. 16. Num. 1. 2000.

[73] *Email Security-Usability Dichotomy : Necessary Antinomy or Potential Synergism?* Vernon M Neppe MD, PhD, FRSSAfa. Available sur : http ://www.pni.org/ethics/security/UsableSecureEmailNeppe2008.pdf

[74] Enge, A. *Elliptic Curve And Their Application To Cryptograpy : An introduction*. Kluwer Academic Publishers, 1999.

[75] Fan, J., Vercauteren, F., Verbauwhede, I. *Efficient Hardware Implementation of Fp-Arithmetic for Pairing-Friendly Curves*. Computers, IEEE Transactions. Vol. 61. Num. 5. pp. 676-685. 2012

[76] Fijusiki, E., Okamoto, T., piontcheval, D., stern J. *RSA-OAEP is Secure under the RSA Assymption*. Journal of Cryptology. Vol. 17. Num. 2. pp. 81-104. Springer-Verlag. 2004.

[77] Fouque, P.A., Kunz-Jacques, S., Martinet, G. Frédéric, M, and Frédéric V. *Power Attack on Small RSA Public Exponent*. Cryptography Hardware and Enbeded Systems-CHES2006. Vol. 4249 of Lecture Notes in Computer Science. Springer Verlag. pp. 339-353. 2006. Available aussi sur : WWW.iacr.org/archives/ches2006/27/27.pdf

[78] Frey, G., Rück, H. *A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves*. Math. Comput. Vol. 62. pp. 865-874. 1994.

[79] Freeman, D., Scott, M., Teske, E. *A taxonomy of pairing friendly elliptic curves*. Cryptology ePrint Archive, Report 2006/372, 2006. Available sur : http ://eprint.iacr.org/.

[80] Freeman, D. *Constructing pairing-friendly elliptic curves with embedding degree 10*. Dans Algorithmic Number Theory Symposium ANTS-VII. Vol. 4076 of Lecture Notes in Computer Science. Springer Verlag (Springer, Berlin, 2006), pp. 452-465. 2006.

[81] Fujisaki, E., Okamoto, T. *How to enhance the security of public-key encryption at minimum cost*. Dans Public Key Cryptography (PKC). Lecture Notes in Computer Science. Springer Verlag. pp. 53-68. 1999.

[82] Fujisaki, E., Okamoto, T. *Secure integration of asymmetric and symmetric encryption schemes*. Dans Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology. Lecture Notes in Computer Science. Springer Verlag. pp. 537-554. 1999.

[83] Galbraith, S., Paterson, K., Smart, N. *Pairings for cryptographers*. Discrete Applied Mathematics. Vol. 156. Num. 16. pp. 3113-3121. 2008.

[84] Galbraith, S., McKee, J., Valença, P. *Ordinary abelian varieties having small embedding degree*. Finite Fields Appl. Vol. 13. pp. 800-814. 2007.

[85] Galbraith, S., Hess, F., Vercauteren, F. *Aspects of pairing inversion*. IEEE Transactions on Information Theory. Vol. 54. Num. 12. pp. 5719-5728. 2008.

[86] Galindo, D. *Boneh-Franklin identity based encryption revisited*. Dans Proceedings of the 32nd International Colloquium on Automata. ICALP. 2005.

[87] Galindo, D. *A separation between selective and full-identity security notions for identity-based encryption*. Available sur : http ://www.cs.ru.nl/paw/publications/separationresultfinal.pdf

[88] Galindo, D,. Hasuo, I. *Security Notions for Identity Based Encryption*. Available sur : http ://eprint.iacr.org/2005/253.

[89] Gary L. Miller. *L'hypothèse et de tests pour Primality*. Journal du système informatique et sciences. Vol. 13. Num. 3. pp. 300-317. 1976.

[90] Gentry, C., Silverberg, A. *Hierarchical ID-based cryptography*. In Yuliang Zheng, editor, Advances in Cryptology - ASIACRYPT 2002. Vol. 2501 of Vol. 2612 of Lecture Notes in Computer Science. Springer Verlag. pp. 548-566. 2002.

[91] Gentry, C. *Practical identity-based encryption without random oracles*. Dans Serge Vaudenay, Editor, Advances in Cryptology - EUROCRYPT 2006. Vol 4004 of Lecture Notes in Computer Science. Springer Verlag. pp. 445-464. 2006.

[92] Gentry, C. *Certificate-based encryption and the certificate revocation problem*. Dans EUROCRYPT 2003. Lecture Notes in Computer Science. Springer Verlag. pp. 272-293. 2003.

[93] Gerck, E. *Secure Email Technologies X.509/PKI, PGP, IBE and Zmail*. Dans Corporate Email Management. Chapter 12, edit par Krishna SJ, Raju E. Hyderabad, India, ICFAI University Press. pp. 171-196. 2007.

[94] Ghosh, S., Mukhopadhyay, D., Chowdhury, D.R. *Fault Attack, Countermeasures on Pairing Based Cryptography*. I. J. Network Security. Vol. 12. Num. 1. pp. 21-28. 2011.

[95] Girault, M,. *Self-certiffied public keys*. Dans Advances in Cryptology, Proceedings of EuroCrypt91. Vol. 547 of Lecture Notes in Computer Science. Springer Verlag. pp. 490-497. 1991

[96] Granger, R., Page, D., Smart, N. P. *High security pairing based cryptography revisited*. Dans Florian Hess, Sebastian Pauli, and Michael E. Pohst, Editors, ANTS VI. Vol. 4076 of of Lecture Notes in Computer Science. Springer Verlag. pp. 480-494. 2006.

[97] Granlund, T et al. *GMP : GNU multiple precision arithmetic library*. Available sur : http ://gmplib.org.

[98] Halevi, Shai. *A sufficient condition for key-privacy*. IBM T.J. Watson Research Center, January 7, 2005.

[99] Hayashi, R., Tanaka, K. *Universally Anonymizable Public-Key Encryption*. Dans Advences in Cryptologu, ASIACRYPT'05. Lecture Notes in Computer Science. Springer Verlag. 2005.

[100] Hess, F. *Efficient Identity Based Signature Schemes Based on Pairings*. Dans Selected Areas in Cryptography-SAC02. Vol. 2595 of Lecture Notes in Computer Science. Springer Verlag. pp. 310-324. 2003.

[101] Hess, F. *Pairing Lattices*. Dans Steven Galbraith and Kenny Peterson, Editors, Pairing 2008. Vol. 5209 of Lecture Notes in Computer Science. Springer Verlag. pp. 18-38, Berlin, Heidelberg, 2008.

[102] Jason Hinek, M. *On the Security of Multi-prime RSA*. Cheriton June 13. 2006. Available sur : http : cacr.math.uwaterloo.ca / techreports / 2006 / cacr2006-16.

[103] Hoang, T. *Application des Couplage En Cryptographie*. 12 July 2005. Available sur : http ://www.math.ucla.edu/ leth/stuff/couplage.pdf.

[104] Horwitz, J., Lynn, B. *Toward hierarchical identity-based encryption*. Dans Lars R. Knudsen, Editor. Advances in Cryptology - EUROCRYPT 2002. Vol 2332 of of Lecture Notes in Computer Science. Springer Verlag. pp. 466-481. 2002.

[105] Huhnlein, D., Jacobson, M.,Weber, D. *Towards practical non-interactive public key cryptosystems using non-maximal imaginary quadratic orders*. Dans SAC 00. Vol 2012 of of Lecture Notes in Computer Science. Springer Verlag. pp. 275-287. 2000.

[106] Icart, T. *How to hash into elliptic curves*. Dans Shai Halevi, Editor, CRYPTO. Vol. 5677 of Lecture Notes in Computer Science. Springer Verlag. pp. 303-316. 2009.

[107] IEEE P1363.3 Committee. *IEEE 1363.3 - standard for identity-based cryptographic techniques using pairings*. Available sur : http ://grouper.ieee.org/groups/1363/, April 2007.

[108] Izabachène, M. *L'anonymat dans les protocoles Cryptographiques 2009*. École Normale Supérieure. Thèse available sur : http ://www.lsv.ens-cachan.fr/ izabache/download/these m.pdf

[109] Izu T., Takagi, T. *Efficient Computations of the Tate Pairing for the Large MOV Degrees*. Dans ICISC 2002. Vol. 25-87 of of Lecture Notes in Computer Science. Springer Verlag. pp. 283-297. 2003.

[110] Jahid, S., Borisov, N. *PIRATTE : Proxy-based Immediate Revocation of ATTribute-based Encryption*. CoRR abs/1208.4877. 2012.

[111] Joux, A. *A one round protocol for tripartite Diffie Hellman*. Dans Algorithmic Number Theory SymposiumANTS-IV. Vol. 1838 of Lecture Notes in Computer Science. Springer Verlag. (Springer, Berlin, 2000). pp. 385-393. Full version : Journal of Cryptology. Vol. 17. pp. 263-276. 2004.

[112] Joye, M., Neven, G (Editors). *Identity-Based Cryptography*. Bouquin. Vol. 2. Cryptology and Information Security Series. IOS Press, Amsterdam, The Netherlands, The Netherlands, 2008. (chapitre de livre).

[113] Junod, P. *Cryptographic Secure Pseudo-Random Bits Generation : the Blum-Blum-Shub Generator*. 1999.

[114] Kachisa, E,J., Schaefer, E, F., Scott, M. *Constructing Brezing Weng pairing friendly elliptic curves using elements in the cyclotomic field*. Pairing-Based Cryptography :Pairing08. Vol. 5209 of Lecture Notes in Computer Science. Springer Verglas. (September 1-3, 2008). pp. 126-135. 2008. DOI. 10.1007/978-3- 540-85538-5_9.

[115] Kantarcioglu, M. *Semantic Security of RSA*. 4 / 1 / 2008. Available on : http : www. utdallas.edu/muratk/ courses /crypto09s files/semantic-rsa.

[116] Katz, J., Wang, N. *Efficiency improvements for signature schemes with tight security reductions*. Dans ACM Conference on Computer and Communications Security - ACM CCS 2003. pp. 155-164. ACM Press. 1993.

[117] Kate, A., Goldberg, I. *Distributed Private-Key Generators for Identity-Based Cryptography*. SCN 2010 : 436-453 Conference : Security and Cryptography for Networks-SCN. pp. 436-453. 2010. DOI : 10.1007/978-3-642-15317-4 27. Aussi available sur : http ://eprint.iacr.org/2009/355.pdf

[118] Kiltz, E., Vahlis, Y. *CCA2 Secure IBE : Standard Model Efficiency through Authenticated Symmetric Encryption*. Dans CT-RSA 08. Lecture Notes in Computer Science. Springer Verlag. T. Malkin, Editor. 2008. Available sur : http ://eprint.iacr.org/2008/020.pdf

[119] Kim, T. H., Takagi, T., Han, D.G., Kim, H. W., Lim, J. *Side Channel Attacks and Countermeasures on Pairing Based Cryptosystems over Binary Fields*. CANS'06. Lecture Notes in Computer Science. Springer Verlag. Proceeding of the 5th conference on cryptology and Network Security. pp. 168-181. 2006.

[120] Klima, v. *Tunnels in Hash Functions : MD5 Collisions Within a Minute*. Cryptology ePrint Archive, Report 2006/105, 2006. Available sur : http ://eprint.iacr.org/.

[121] Koblitz, N. *Elliptic curve cryptosystems*. Mathematics of Computation. Vol. 48. pp. 203-209. 1987.

[122] Koblitz, N,. Menezes, A. *Pairing-based cryptography at high security levels*. Dans Nigel P. Smart, Editor. Cryptography and Coding, Vol. 3796 of Lecture Notes in Computer Science. Springer Verlag. pp. 13-36, Berlin, Heidelberg, 2005.

[123] Kocher, C., Jaffe, J., Jun, B. *Differential Power Analysis*. CRYPTO99. Vol. 1666 of Lecture Notes in Computer Science. Springer Verlag. pp. 388-397, 1999.

[124] Kumar, K, P., Shailaja, G., Saxena, A. *Secure and Efficient Threshold Key Issuing Protocol for ID-based Cryptosystems*. Cryptology ePrint Archive, Report 2006/245. Available sur : http ://eprint.iacr.org/2006/245.pdf

[125] Leurent, G., Nguyen, P. Q. . *How risky is the random-oracle model ?* Dans Halevi, Editors. Advances in Cryptology - CRYPTO09 Lecture Notes in Computer Science. Springer Verlag. Vol. 5677. pp 445-464. 2009.

[126] Libert B,. *Quisquater J J. Efficient revocation and threshold pairing based cryptosystems*. Dans Proceeding 22nd ACM Symposium on Principles of Distributed Computing (PODC03). Boston, USA, July 13-16, 2003, pp. 163-171. 2003.

[127] Lidl, R., Niederreiter, H. *Finite Fields*. Cambridge University Press, Cambridge, book. Second Edition. 1994.

[128] Lim, H. W. *On the Application of Identity-Based Cryptography in Grid Security*. Royal Holloway, University of London, 2006. Thèse available sur : http ://www.isg.rhul.ac.uk/ kp/theses/HWLthesis.pdf

[129] Liu, J. K., Ho Au, M., Willy, S. *Self-Generated-Certificate Public Key Cryptography and Certificateless Signature / Encryption Scheme in the Standard Model*. ACM Symposium on Information, Computer and Communications Security (ASIACCS 2007). pp. 273 - 283. ACM Press. 2007.

[130] Lynn, B. *ON The Implementation Of Paring-Based Cryptosystems*. Departement Of Computer Science And The Comettee. Stanford University. Thèse. 2007.

[131] *MAGMA Computational Algebra System. MAGMA version V2.19-4, 2013*. Available sur : http ://magma.maths.usyd.edu.au/magma.

[132] Martin, L. *Introduction To Identity Based Encryption*. Artech Home INC. 2008.

[133] Maurer, U. Yacobi, Y. *Non-interactive public-key cryptography*. Dans Crypto 91. Vol. 547 of Lecture Notes in Computer Science. Springer Verlag. pp. 498-507. 1991.

[134] Menezes, A., Okamoto, T., Vanstone, S. *Reducing elliptic curve logarithms to logarithms in a finite field*. IEEE Trans. Inf. Theory. Vol. 39. pp. 1639-1646. 1993.

[135] Miller, V. *Use of elliptic curves in cryptography*. Advances in cryptology CRYPTO 85. 218 of Lecture Notes in Computer Science. Springer Verlag. pp. 417-426. New York, NY, USA. New York, Inc. 1986.

[136] Miller, V. S. *Short programs for functions on curves*. unpublished, 1986. Available sur : http ://crypto.stanford.edu/miller/miller.pdf.

[137] Miller, V. S. *The Weil pairing, and its efficient calculation*. Dans Jounal of Cryptology. Vol. 17. pp. 235-261. Springer-Verlag. Secaucus, NJ, USA. Springer-Verlag. 2004.

[138] Mitsunari, S., Saka, R., Kasahara, M. *A new traitor tracing*. IEICE Transactions. E85-A(2). pp. 481-484. Feb. 2002.

[139] Miyaji, A., Nakabayashi, M., Takano, S. *New explicit conditions of elliptic curve traces for FRreduction*. IEICE Trans. Fundam. E84-A(5), pp. 1234-1243. 2001.

[140] Naccache, D. *Secure and Practical Identity-Based Encryption*. Available sur : http ://eprint.iacr.org/2005/369.pdf

[141] Naor, M., Pinkas, B. *Efficient Trace and Revoke Schemes*. Dans FC, 2001. 4th International Conference, FC 2000 Anguilla, British West Indies, February 20-24, 2000 Proceedings. pp. 1-20. DOI :10.1007/3-540-45472-1 1

[142] Nitaj, A. *CRYPTANALYSE DE RSA*. Available sur : http ://www.math.unicaen.fr/ nitaj/CryptRSA.pdf

[143] Page, D., Vercauteren, F. *Fault and side channel attacks on pairing based cryptography*. Dans IEEE Transactions on Computers. 2006.

[144] Paterson, K. G. *ID-based signatures from pairings on elliptic curves*. Cryptology ePrint Archive, Report 2002/004, 2002. Available sur : http ://eprint.iacr.org/2002/004.

[145] Paterson, K. *Pairing-Based Cryptography An Introduction*. ECRYPT Summer School Samos. May 4th 2007

[146] Phan, D, H. *Sécurité et efficacité des schémas cryptographiques*. École Normale Supérieure. École polytechnique Département d'informatique. Thèse. 2005.

[147] Pirretti, M., Traynor, P., McDaniel, P.,Waters, B. *Secure Atrribute-Based Systems*. ACM conference on Computer and Communications Security (ACM CCS). 2006.

[148] Pulkkis, G., Grahn, K., Karlsson, J. *Some Implementation Issues for Security Services based on IBE*. Proceedings CONF-IRM. 2008

[149] Qing-hai, B. *Comparative research on two kinds of certification systems of the public key infrastructure (PKI) and the identity based encryption (IBE)*. Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC). pp. 147 - 150. 2012.

[150] Rabin, M, O. *Algorithme probabiliste de primalité d'essai*. Journal of Number Theory. Vol. 12. Num. 1. pp. 128-138, DOI. 10.1016/0022-314X (80) 90084-0. 1980

[151] Rivest, R., Shamir, A. Adleman, L. *A method for obtaining digital signatures and public-key cryptosystems*. Communications of the ACM. Vol. 21. Num. 2. pp. 120-126. 1978.

[152] RSA Laboratories. *RSAES-OAEP Encryption Scheme. RSA Security Inc*. 20 Crosby Drive Bedford, MA 01730 USA.

[153] Sahai, A., Waters, B. *Fuzzy Identity Based Encryption*. Advances in Cryptology Eurocrypt. Vol. 3494 of Lecture Notes in Computer Science. Springer Verlag. pp. 457-473. 2005.

[154] Sakai, R., Kasahara, M. *Id based cryptosystems with pairing on elliptic curve*. Cryptology ePrint Archive, Report 2003/054, 2003. http ://eprint.iacr.org/.

[155] Sakai, R., Ohgishi, K., Kasahara, M. *Cryptosysytems based on pairing*. Proceedings of Symposium on Cryptography and Information Security. SCIS00. 2000.

[156] Sanjit C,. Sarkar, Palash. *Constant Size Ciphertext HIBE in the Augmented Selective-ID Model and its Extensions*. IACR eprint archive report 084/2007.

[157] Satoh, T. *On pairing inversion problems*. T. Takagi, E. Okamoto, and T. Okamoto, Editors, Pairing 2007. Vol. 4575 of Lecture Notes in Computer Science. Springer Verlag. pp. 317-328. Berlin, 2007.

[158] Schoof, R. *Elliptic curves over finite field and the computation of square roots mod p*. Mathematics of Computation. Vol. 44. pp. 483- 494. 1985

[159] Scott, M. *A Note on Boneh and Franklin IBE*. Available sur : ftp ://ftp.computing.dcu.ie/pub/resources/crypto/note.pdf

[160] Scott, M., Barreto. P. S. L. M. *Generating more MNT elliptic curves. Designs, Codes and Cryptography*. Vol. 38. Num. 2 (Kluwer Academic Publishers Norwell, MA, USA, February 2006). pp. 209-217. 2006. DOI. 10.1007/s10623- 005-0538-1.

[161] Shamir, A., *Identity Based cryptosystems and signature shemes*. Advances in cryptology-crypto84. Vol. 196 of Lecture Notes in Computer Science. Springer Verlag. pp. 47-53. 1984.

[162] Shoup, V. *OAEP Reconsidered*. Dans Crypto 2001. Vol. 2139 of Lecture Notes in Computer Science. Springer Verlag. pp. 239-259. Berlin,2001.

[163] Shoup, V. *NTL : A library for doing number theory*. http ://www.shoup.net/ntl.

[164] Silverman, J. H. *The arithmetic of elliptic curves*. Vol. 106 of Graduate Texts in Mathematics. Springer-Verlag, New York. 1992. Corrected reprint of the 1986 original.

[165] Sugita, Makoto., Kawazoe, M., Perret, L., Imai, H. *Algebraic Cryptanalysis of 58-Round SHA-1*. Alex Biryukov, Editor, FSE. Vol. 4593 of of Lecture Notes in Computer Science. Springer Verlag. pp. 349-365. 2007.

[166] Sun, Y., Zhang, Futai., Baek, J. *Strongly Secure Certificateless Public Key Encryption Without Pairing*. In Bao,F ,Ling,S ,Okamoto,T,Wang, H, Xing, C, Editors. CANS 2007. Vol. 4856 of Lecture Notes in Computer Science. Springer Verlag. pp. 194-208, Heidelberg. 2007.

[167] SY Yan. *Cryptanalytic attacks on RSA*. 2008 - lavoisier.fr.

[168] Tanaka, N., Saito, T.*On the q-Strong Diffie-Hellman Problem*. IACR eprint archive report 215/2010 http ://eprint.iacr.org/2010/215.pdf

[169] Tanaka, H. *A realization scheme for the identity-based cryptosystem*. Dans Crypto87. Vol. 293 of Lecture Notes in Computer Science. Springer Verlag. pp. 341-349. 1987.

[170] Tsuji, S., Itoh T. *An ID-based cryptosystem based on the discrete logarithm problem*. IEEE Journal on Selected Areas in Communication. Vol. 7. Num. 4. pp. 467-473. 1989.

[171] Vercauteren, F. *Optimal Pairings*. Cryptology ePrint Archive, Report 2008/096. Available sur : http ://eprint.iacr.org/2008/096, 2008.

[172] Verheul, E. R. *Evidence that XTR Is More Secure than Supersingular Elliptic Curve Cryptosystems*. Journal of Cryptology. Vol. 17. pp. 277 - 296. 2004.

[173] Waters, B. *Efficient identity-based encryption without random oracles*. Dans Ronald Cramer, Editor, Advances in Cryptology - EUROCRYPT05. Vol. 3494 of Lecture Notes in Computer Science. Springer Verlag. pp. 114-127. 2005.

[174] Whelan, C., Scott, M. *Side channel analysis of practical pairing implementation : Which path is more secure*. Dans VietCrypt 2006, of of Lecture Notes in Computer Science. Springer Verlag. 2006.

[175] Yunqi, D., Jiang, W., Yun, W., Chuangui, Ma. *Fault Attack Against Miller's Algorithm for Even Embedding Degree*. International Journal of Network Security. Vol.16. Num.3. pp. 185-193. May 2014.

[176] Zhao. C.-A., Zhang, F. *Reducing the Complexity of the Weil Pairing Computation*. Cryptology ePrint Archive, Report 2008/212, 2008. http ://eprint.iacr.org/2008/212.

[177] http ://csrc.nist.gov/groups/ST/hash/sha-3/Round3/index.html