

N° d'ordre : 3477

THESE

En vue de l'obtention du : **DOCTORAT**

Centre de Recherche : Centre de Recherches Mathématiques et Applications de Rabat

Structure de Recherche : Laboratoire de mathématiques, informatique et applications - sécurité de l'information

Discipline : Informatique

Spécialité : Cryptographie et Sécurité de l'information

Présentée et soutenue le 15/06/2021 par :

Edoukou Berenger AYEIE

New Post-quantum Cryptographic Primitives Schemes

JURY

Mme Mouline SALMA	PES, Université Mohammed V-Rabat, Faculté des Sciences	Présidente
M. Thierry BERGER	PES, Université de Limoges - Limoges	Rapporteur
M. Sidi Mohamed DOURI	PH, Université Mohammed V-Rabat, Faculté des Sciences	Rapporteur/Examineur
M. Mohammed ERRADI	PES, Université Mohammed V-Rabat, École Nationale Supérieure d'Informatique et d'Analyse des Systèmes	Rapporteur/Examineur
M. Christophe PETIT	PES, Université Libre de Bruxelles - Bruxelles	Rapporteur/Examineur
M. Oussama Mohamed REDA	PH, Université Mohammed V-Rabat, Faculté des Sciences	Examineur
M. El Mamoun SOUIDI	PES, Université Mohammed V-Rabat, Faculté des Sciences	Directeur de Thèse

Année Universitaire : 2020/2021

” *"La science, dans ses résultats, est plus magique que
la magie: c'est une magie à preuves!"*

— **Jean-Marie Adiaffi**
La Carte d'identité

A mon père et à ma mère...

A mes frères et à ma soeur...

A la mémoire de mes grands parents...

Acknowledgement

Les travaux présentés dans ce manuscrit ont été effectués au sein du Laboratoire de Mathématiques, Informatique et Applications-Système d'information (LabMIA-SI) de l'Université Mohammed V-Rabat, faculté des sciences, sous la direction du Professeur El Mamoun SOUIDI.

Au terme de ce doctorat, j'adresse toute mon estime à chaque personne qui a contribué de près ou de loin à sa réalisation. Ainsi j'adresse mes sincères remerciements:

A Monsieur le Professeur El Mamoun SOUIDI, Directeur de ma thèse et Professeur de l'Enseignement Supérieur à la Faculté des Sciences, Université Mohammed V-Rabat, qui m'a d'abord accueilli dans son Master spécialisé en Codes, Cryptographie et Sécurité de l'information, puis au sein du Laboratoire de mathématiques, informatique et applications - sécurité de l'information, puis a consenti à assurer mon encadrement pour le Master et aussi pour le Doctorat. Je vous suis extrêmement reconnaissant pour les efforts que vous déployés tout au long de mon cursus universitaire et plus particulièrement pour l'aboutissement de ma thèse. Merci de m'avoir fait bénéficier de vos impressionnantes connaissances scientifiques et d'un encadrement d'exception. Merci pour vos conseils avisés, vos encouragements afin que je puisse arriver au bout de ce travail de thèse, de même que votre écoute et votre disponibilité permanente.

A Madame Mouline SALMA, Professeur de l'Enseignement Supérieur à la Faculté des Sciences, Université Mohammed V-Rabat et Coordinatrice du projet de recherche "Smart Guide : Recommandations intelligentes d'offres touristiques par exploitation des Big Data", pour avoir très aimablement accepté de présider le jury pour la soutenance de ma thèse malgré les différentes occupations liées à ses fonctions.

J'adresse également mes vifs remerciements à Monsieur Thierry BERGER, Professeur de l'Enseignement Supérieur à l'Université de Limoges, pour avoir très aimablement accepté d'évaluer et juger ce travail en tant que rapporteur, malgré vos occupations.

J'aimerais également remercier Monsieur Sidi Mohamed DOUIRI, Professeur Habilité à la Faculté des Sciences, Université Mohammed V-Rabat, pour m'avoir honoré de sa présence en acceptant d'être rapporteur et examinateur de ce travail. Vous avez accepté de prendre de votre temps pour juger cette thèse. C'est une grande joie de vous compter parmi les membres du jury.

Mes sincères remerciements s'adressent également à Monsieur Mohammed ERRADI, Professeur de l'Enseignement Supérieur à l'École Nationale Supérieure d'Informatique et d'Analyse des Systèmes (ENSIAS), pour m'avoir honoré de sa présence en acceptant d'être rapporteur et examinateur de ce travail. C'est un honneur de vous compter parmi les membres du jury.

Je tiens aussi à remercier Monsieur Christophe PETIT, Professeur à l'Université libre de Bruxelles, pour avoir accepté d'être rapporteur de ce travail. C'est un honneur de vous compter parmi les membres du jury.

Qu'il me soit permis de remercier vivement Monsieur Oussama Mohamed REDA, Professeur habilité à la Faculté des Sciences, Université Mohammed V-Rabat. Vous avez eu la gentillesse d'accepter d'être examinateur. Je tiens à vous faire part de ma respectueuse gratitude et de ma joie.

Je ne saurais remercier ici les personnes dont la collaboration a été essentielle pour la réalisation de certaines étapes de ce travail, en particulier toute l'équipe du Laboratoire de mathématiques, informatique et applications - sécurité de l'information pour leur accueil à chaque fois chaleureux et amical. Je remercie très sincèrement mon ami, mon ancien de tous les jours, Dr. Moussa TRAORE pour ses conseils et surtout de toujours répondre présent à chacun de mes appels à l'aide. Je ne veux pas laisser passer cette occasion sans remercier mes amis qui sont comme une deuxième famille pour moi ; ceux qui étaient à mes côtés dès le début notamment DEMBELE Boua Siriki et DIABY Abdoul Razack; ceux qui partagent mon quotidien, FOFANA Valy, Dr. KANGA Junior, Dr. ASSEMIAN Ines, SORO Elisabeth, Dr. Raymond ALLOU, Irna DJOHAR, TRAORE Halassane Falikou, WAM Germain, DIASSO Moussa et tous les autres si par oubli j'aurais oublié de mentionner votre nom. Je remercie, honnêtement toute personne qui a participé de près ou de loin, directement ou indirectement à la réalisation de ce travail.

Enfin, mes parents, mes grands-parents, mes frères et sur, mes cousins et cousines, mes oncles et tantes, mes neveux et nièces et tous mes amis de leur soutien et amour.

Abstract

The sum of this computer and the Shor algorithm will compromise the security of many commonly used cryptographic algorithms based on number theory. Then, many researchers have begun to investigate to develop cryptographic algorithms that would be secure against quantum computers. In this thesis we develop a new code-based cryptographic primitives schemes. Our contribution to code-based cryptography starts with showing that it is possible to reduce the public key length of the first provably secure group signature scheme from code-based assumptions without losing the security properties. Thereafter, we propose a new dynamic code-based group signature without trapdoor matrix. The general idea of this scheme consists in finding a collision between two ciphertexts using two different QC-MDPC matrices in McEliece cryptosystem. Our contribution continues by using random double circulant codes to design a rank metric version of AGS identification scheme which is a five pass protocol with a cheating probability of $\frac{1}{2}$. Another way of our contribution is the presentation of the first mutual authentication protocol for RFID systems based on difficult problems from rank metric codes. We finish our contribution with a proposition of a cryptographic accumulator scheme and a fully dynamic group signature scheme. We instantiate our protocol with a double circulant code which allows us to get a small key size.

Keywords Error Correcting Codes; Post-Quantum Cryptography; Zero-knowledge Identification Scheme; Group Signature; Cryptographic Accumulator; Tag RFID

Résumé

L'ordinateur quantique associé à l'algorithme de Shor pourrait compromettre la sécurité de la majorité des algorithmes cryptographiques à base de théorie des nombres. Pour parer à cette menace, des recherches ont été entreprises en vue de développer des nouveaux algorithmes cryptographiques, appelés cryptographie post-quantique, qui seraient sûrs face à l'ordinateur quantique. Dans cette thèse nous nous sommes attelés à développer de nouveaux schémas en utilisant des primitives cryptographiques à base de codes correcteurs d'erreurs. Nous avons débuté nos travaux en montrant qu'il était possible de réduire la taille de la clé publique du premier schéma de signature de groupe prouvé sûr à base de codes correcteurs d'erreurs tout en conservant les propriétés de sécurité. Ensuite, sans utiliser de matrice à trappe, nous avons proposé un nouveau schéma de signature de groupe dynamique à base de codes correcteurs d'erreurs. L'idée générale de ce schéma est de trouver une collision entre deux textes chiffrés par deux différentes instances de la version QC-MDPC du cryptosystème de McEliece. Nos travaux s'enchaînent avec l'utilisation de codes aléatoires doublements circulants pour proposer un schéma d'identification à divulgation d'informations nulles cinq-passes avec une probabilité de triche de $\frac{1}{2}$. Dans ces travaux, nous avons aussi proposé le premier schéma d'authentification mutuelle pour les tags RFID basé sur les codes en métrique rang. Nos travaux se terminent par la proposition d'un schéma d'accumulateur cryptographique que nous avons utilisé pour proposer un schéma de signature totalement dynamique basé sur les codes correcteurs d'erreurs.

Mot-clés Codes Correcteurs d'Erreurs, Cryptographie Post-Quantique, Protocole à Divulgation d'informations nulle, Schéma de signature de groupe, Accumulateur Cryptographique.

Résumé étendu

Ces dernières années ont été le théâtre d'une recherche acharnée dans le domaine l'ordinateur quantique. Ces recherches ont vu naître l'ordinateur quantique qui associé à l'algorithme de Shor pourrait compromettre la sécurité de la majorité des algorithmes cryptographiques à base de théorie des nombres et de courbes elliptiques. Pour parer a cette menace, des recherches ont été entreprises en vue de développer des nouveaux algorithmes cryptographiques, appelés cryptographie post-quantique, qui seraient sûr face à l'ordinateur quantique. Il existe plusieurs candidats à la cryptographie post-quantique, les plus connus sont celle basées sur les codes correcteurs d'erreurs, les réseaux euclidiens, les isogénies entre courbes elliptiques et celles basées sur les polynômes multivariés. La rédaction de cette thèse sest faite au tour des chapitres suivants : le chapitre 1 a été consacré à l'introduction générale, ensuite dans le chapitre 2 nous avons fait les rappels sur les codes correcteurs derreurs et des primitives cryptographiques nécessaire à la compréhension de la thèse. Dans Le chapitre 3, nous avons montré qu'il était possible de réduire la taille de la clé publique du premier schéma de signature de groupe prouvé sûr à base de codes correcteurs d'erreurs tout en conservant les propriétés de sécurités. Ensuite, dans le chapitre 4, sans utilisé de matrice à trappe, nous avons proposé un nouveau schéma de signature de groupe dynamique à base de codes correcteurs d'erreurs. Nos travaux s'enchainent avec le chapitre 5 où il est question d'utiliser les codes aléatoires doublements circulants pour proposer un schéma d'identification à divulgation dinformations nulles cinq-passes avec une probabilité de triche de $1/2$ qui est une version en métrique rang du schéma d'identification ags. Le chapitre 6 a quant à lui été dédié à la proposition du premier schéma d'authentification mutuelle pour les tags RFID basé sur les codes en métrique rang. Ensuite dans le chapitre 7, nous proposons un schéma d'accumulateur cryptographique utilisé pour proposer un schéma de signature totalement dynamique basé sur les codes correcteurs d'erreurs. Enfin dans le dernier chapitre, nous faisons la conclusion générale de notre thèse.

Dans le domaine des codes correcteurs derreurs il existe deux schémas de signature de groupe, le second est un schémas de signature de groupe statique proposé par Ezerman et al. Ce premier schéma de signature de groupe basé sur les code correcteurs derreur prouvé sûr proposé par Ezerman et al a été proposé sans tenir compte de la taille des clé qui sont très grande, ce qui rend

ce schéma impraticable en situation réelle. Dans le chapitre 3 nous montrons qu'il est possible de réduire la taille de la clé publique premier schéma de signature de groupe basé sur les codes correcteurs de Reed-Solomon prouvé sûr sans pour autant perdre les propriétés de sécurité. En effet, en utilisant une variante du cryptosystème McEliece qui utilise les codes ayant leur matrices de contrôle à densité modérée quasi-circulantes pour le chiffrement de l'identité du signataire et les codes doublements circulants aléatoires pour le système d'argument à divulgation d'information nulle nous avons obtenu une clé publique 466 fois plus petite que dans le schéma original. Typiquement, pour un groupe de 16 personnes nous obtenons une clé publique de 1.34 Kilo-bits tandis que dans le schéma original la clé publique obtenue est de 625 Kilo-bits. Notons que tous les schémas de signatures de groupes dynamiques post-quantiques utilisent des matrices à trappe. Ces matrices à trappe représentent un gros problème vu qu'elles sont difficiles à trouver en pratique et le seul candidat à ce jour reste la matrice CFS et qui n'est pas efficace en pratique. Dans le chapitre 5 nous proposons un nouveau schéma de signature de groupe dynamique basée sur les codes correcteurs de Reed-Solomon sans utiliser de matrice à trappe. L'idée générale ici consiste à créer une collision entre deux textes chiffrés avec deux instances de la version du cryptosystème McEliece qui utilise les codes ayant leur matrices de contrôle à densité modérée quasi-circulantes. Au lieu d'utiliser la version originale d'AGS, qui a comme clé privée (e, m) tel que le poids $w(e) < t$ la capacité de correction et clé publique (G, x, t) avec $x = mG + e$, nous avons utilisé une version modifiée de AGS, dans cette version nous avons utilisé deux instances d'AGS afin d'éviter de conserver dans une matrice les clés publiques des utilisateurs. Pour créer une collision et passer outre les matrices à trappe, nous avons utilisé les codes QC-MDPC. Cette nouvelle version d'AGS a pour clé privée (e, m) tel que le poids $w(e) < t$ et $mG + e = 0$ et clé publique (G, t) . Nous utilisons aussi une variante du schéma d'identification à divulgation d'informations nulle AGS pour prouver la possession de la clé publique par le signataire, ensuite nous utilisons la transformation CFS pour obtenir une signature de groupe. Dans le schéma proposé ici, la taille des clés et de la signature reste constantes et indépendantes de la taille du groupe et restent plus petites que celles proposées dans la littérature. En 1995 Chen a proposé un nouveau schéma d'identification à divulgation d'informations nulle basé sur les codes correcteurs de Reed-Solomon en utilisant une autre métrique appelée métrique de rang avec une probabilité de triche de $2/3$ mais avec une clé publique de grande taille. Dans ce chapitre, en utilisant les codes doublements circulants aléatoires, nous concevons une version métrique rang du schéma d'identification AGS qui est un protocole à cinq passes et avec une probabilité de triche de $1/2$. De plus, les matrices utilisées pour masquer les secrets des utilisateurs pendant l'exécution du protocole sont également circulantes. Cette technique combinée à la métrique Rank au lieu de la métrique Hamming fournit un schéma d'identification efficace avec une petite taille de clé et un faible coût de communication. En pratique, notre schéma est plus efficace avec une taille des données publiques de 276 bits au lieu de 350 bits dans le protocole AGS.

pour un niveau de sécurité de 80 bits ce qui implique une réduction de 21%. Aussi, pour un niveau de sécurité de 128 bits, nous obtenons 434 bits comme taille de données publiques au lieu de 550 bits dans le cas du protocole AGS, ce qui implique un gain de 21%. Le chapitre 6 présente le premier protocole d'authentification mutuelle pour les systèmes RFID basé sur des problèmes difficiles de codes correcteurs de erreurs en métriques rang. De plus, notre protocole propose deux grandes améliorations, la première consiste à utiliser des codes ayant leur matrices de contrôle à faible rang doublements circulantes. Cette amélioration permet de réduire les coûts de communication et d'obtenir une taille de clé plus petite que les précédents protocoles à base de codes correcteurs de erreurs. La seconde amélioration consiste en un renforcement de la sécurité en utilisant une fonction de hachage pour obtenir une propriété de sécurité que nous appelons "Full Inside secrecy", ce qui signifie que même si nous avons un oracle qui arrive à corrompre un lecteur contenant des secrets, un adversaire ne peut pas extraire les clés privées des Tags RFID. De plus, si un adversaire peut corrompre un ou plusieurs Tag RFID, il ou elle ne peut pas produire un lecteur qui peut être authentifié par un Tag RFID qui n'a pas été corrompu. De plus, notre protocole économise au moins 42% en termes de données de communication par rapport aux protocoles d'authentification basés sur les codes correcteurs de erreurs pour les systèmes RFID. Un accumulateur cryptographique est une primitive cryptographique qui produit un agrégat succinct d'un ensemble d'éléments. Ce type de schéma permet de produire une preuve d'appartenance pour chaque élément de l'ensemble. En 2019, à l'aide des arbres de Merkle, Nguyen et al. ont proposé un premier accumulateur cryptographique basé sur des codes correcteurs de erreurs. Leur schéma proposé utilise des codes aléatoires qui nécessitent une grande capacité de stockage et ils ne fournissent pas une implémentation de leur schéma. Dans Le chapitre 7, nous proposons un nouvel accumulateur cryptographique basé sur les codes correcteurs de erreurs. Comme dans premier accumulateur cryptographique basé sur des codes correcteurs de erreurs, notre accumulateur cryptographique utilise les arbres de Merkle en utilisant une famille de fonctions de hachage résistantes aux collisions basées sur un code binaire $[2k, k]$. De plus, cette technique utilise des codes doublements circulants qui fournissent des clés compactes et repose de manière sécurisée sur la robustesse du problème de décodage par syndrome. De plus, pour montrer la faisabilité de notre schéma, nous proposons une implémentation de notre accumulateur cryptographique qui est aussi, à notre connaissance, la première implémentation directe d'accumulateurs cryptographiques post-quantiques. En effet, les autres implémentations d'accumulateurs cryptographiques sont déduites des implémentations de primitives cryptographiques qui utilisent les accumulateurs cryptographiques. Pour finir, notre construction permet de déduire un schéma de signature de groupe entièrement dynamique basé sur les codes correcteurs de erreurs qui a une complexité logarithmique en fonction de la taille du groupe.

List of my Publications

1. **Ayebie, E. B.**, and Souidi, E. M.: *Rank-metric code-based mutual authentication protocol for RFID*. In Journal of Information Security and Applications, vol. 55, p. 102-158, 2020. <https://doi.org/10.1016/j.jisa.2020.102598>.
2. **Ayebie, E. B.**, Assidi, H., and Souidi, E. M.: *An Efficient Identification Scheme Based on Rank Metric*. In Foundations and Practice of Security - FPS 2019, Toulouse, France, November 5-7, ser. Lecture Notes in Computer Science, vol. 12056, Springer, 2019, pp. 273-289. https://doi.org/10.1007/978-3-030-45371-8_17.
3. Assidi, H., **Ayebie, E. B.**, and Souidi, E. M.: *An efficient code-based threshold ring signature scheme*. Journal of information security and applications, vol. 45, pp. 52-60, 2019. <https://doi.org/10.1016/j.jisa.2019.01.006>.
4. Belabssir, S., **Ayebie, E. B.**, and Souidi, E. M.: *Perfect, Hamming and Simplex Linear Error-Block Codes with Minimum π -distance 3*. In International Conference on Codes, Cryptology, and Information Security- C2SI 2019, Rabat, Morocco, April 22-24, ser. Lecture Notes in Computer Science, vol. 11445, pp. 288-306. Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-16458-4_17.
5. **Ayebie, E. B.**, Assidi, H. and Souidi, E. M.: *A new dynamic code-based group signature scheme*. In International Conference on Codes, Cryptology, and Information Security- C2SI 2017, Rabat, Morocco, April 10-12, vol. 10194, pp. 346-364, Springer, Cham, 2017. https://doi.org/10.1007/978-3-319-55589-8_23.
6. Assidi, H., **Ayebie, E. B.** and Souidi, E. M.: *Two Mutual Authentication Protocols Based on Zero-Knowledge Proofs for RFID Systems*. In International Conference on Information Security and Cryptology, Lecture Notes in Computer Science, vol. 10779, pp. 267-283, Springer, Cham, 2017. https://doi.org/10.1007/978-3-319-78556-1_15.

7. Assidi, H., **Ayebie, E. B.**, Souidi, E. M.: *A Code-based Group Signature Scheme with Shorter Public Key Length.* In 13th International Joint Conference on e-Business and Telecommunications (ICETE 2016) - Volume 4: SECRIPT, Lisbon, Portugal, July 26-28, SciTePress, 2016, pp. 432439. <https://doi.org/10.5220/0005969204320439>.
8. **Ayebie, E. B.**, and Souidi, E. M.: *A Code Based Cryptographic Accumulator And Fully Dynamic Group Signature.* Under review (Submitted to Design Codes and Cryptography).

Contents

Acknowledgment	ii
Abstract	iv
List of my Publications	ix
Acronyms	xv
List of Figures	xvii
List of Tables	xviii
List of Algorithms	xix
List of Algorithms	xix
1 Introduction	1
1.1 Historical Cryptography	2
1.2 Modern Cryptography	3
1.2.1 Symmetric Cryptography	3
1.2.2 Asymmetric Cryptography	3
1.2.3 Post-Quantum Cryptography	5
1.3 Thesis Context and Related Works	7
1.4 Manuscript Organisation	14
2 Definitions and Background	16
2.1 Linear Codes	16
2.1.1 Hamming metric	17
2.1.2 Rank metric	23
2.2 Zero-Knowledge Identification Scheme	27
2.3 Group Signature	29
2.3.1 Static Group Signature	29

2.3.2	Dynamic Group Signature	31
2.3.3	Fully Dynamic Group Signature	34
2.4	Cryptographic Accumulator	38
2.5	Conclusion	42
3	A Code-based Group Signature Scheme with Shorter Public Key Length	43
3.1	Introduction	43
3.2	Preliminaries	45
3.2.1	Notations	45
3.2.2	Code-based Cryptography Background	45
3.3	Our Proposed Group Signature Scheme	47
3.4	Security Analysis	50
3.5	Results	52
3.6	Conclusion	52
4	A New Dynamic Code-Based Group Signature Scheme	53
4.1	Introduction	54
4.2	The proposed dynamic group signature	55
4.2.1	Setup algorithm	56
4.2.2	AGS variant protocol with collision	56
4.2.3	High level overview	59
4.2.4	Join scheme	60
4.2.5	Signature algorithm	61
4.2.6	Verification algorithm	61
4.2.7	Open algorithm	62
4.3	Security Analysis and Practical Results	63
4.3.1	Security Analysis	63
4.3.2	Practical results	67
4.4	Conclusion	68
5	An Efficient Identification Scheme Based on Rank Metric	69
5.1	Introduction	69
5.2	The Proposed Zero-knowledge Identification Protocol	72
5.2.1	Key Generation	74
5.2.2	Identification Protocol	74
5.3	Properties and Security of the Scheme RankId	75
5.4	Parameters and Results	81
5.5	Conclusion	83

6	Rank-Metric Code-Based Mutual Authentication Protocol for RFID	84
6.1	Introduction	84
6.2	The Proposed Scheme	87
6.2.1	Description of the Proposed Scheme	88
6.2.2	Security Analysis	89
6.3	Practical Results and Comparison	93
6.4	Conclusion	96
7	A Code Based Cryptographic Accumulator and Fully Dynamic Group Signature	97
7.1	Introduction	98
7.2	Preliminaries	100
7.2.1	Code-based Cryptography	100
7.2.2	Zero-Knowledge Argument Systems and Stern-Like Protocols	101
7.2.3	Collision-Resistant Hash Function	102
7.3	The Proposed Code Based Cryptographic Accumulator	103
7.3.1	A Code Based Difficult Problems	104
7.3.2	A Family of Code-Based Collision-Resistant Hash Functions	105
7.3.3	The Proposed Code-based Accumulator	107
7.3.4	Example	109
7.4	Security and Implementation of the Proposed Cryptographic Accumulator	111
7.4.1	Security of the Proposed Cryptographic Accumulator	111
7.4.2	Implementation	115
7.5	The Proposed Fully Dynamic Group Signature scheme	116
7.5.1	Logarithmic-Size Arguments of Set Membership	116
7.5.2	An Update Algorithm	119
7.5.3	Description of Our <i>FDGS</i> Scheme	120
7.5.4	Analysis of Our <i>FDGS</i> Scheme	126
7.5.5	Efficiency	131
7.6	Conclusions	132
	Conclusion and Perspectives	132
	References	137

Acronyms

AES Advanced Encryption Standard. 3

AGS Aguilar, Gaborit and Schrek. iii, iv, xiv, 9, 10, 27, 29

CFS Courtois-Finiasz-Sendrier. 9

CRC Cyclic Redundancy Check. 12

DC-LRPC Double Circulant Low Rank Parity Check. xiii, 26

DDH Decisional Diffie-Hellman. 8

DES Data Encryption Standard. 3

FDGS Fully Dynamic Group Signature. 34, 35, 37

GD General Decoding. 9, 18, 19

LWE Learning with Errors. 6

MDPC Moderate Density Parity-Check. 21, 23

MPKC Multivariate Public-Key Cryptosystem. 6, 7

NIST National Institute of Standards and Technology. 3, 6, 133

PPT Probabilistic Polynomial Time. 30, 31, 37, 41

QC Quasi-Cyclic. 21

QC-MDPC Quasi-cyclic Moderate Density Parity-Check. iii, iv, xiii, 8, 9, 16, 21–23

RFID Radio Frequency Identification. iii, iv, x, 7, 11, 12, 14, 84, 86, 88, 90, 92, 94, 96

RGD Rank General Decoding. 25

RSA Rivest Shamir Adleman. 4, 5, 8

RSD Rank Syndrome Decoding. 25, 26

SD Syndrome Decoding. 18, 19

SIS Short Integer Solutions. 6

List of Figures

1.1	Venn Diagram of Complexity Classes	4
7.1	Merkel Tree Associate to the set $\mathcal{X}$	110

List of Tables

2.1	Parameters and key size for McEliece with Goppa codes [28]	21
2.2	Some proposed Quasi-cyclic Moderate Density Parity-Check (QC-MDPC) instances [105].	23
2.3	Some proposed Double Circulant Low Rank Parity Check (DC-LRPC) instances [68].	26
3.1	Public keys in the original scheme case and in our new scheme case	52
5.1	Comparison between different code-based Zero-knowledge schemes for a 2^{-16} cheating probability with a security of 2^{80}	82
5.2	Comparison between different code-based Zero-knowledge schemes for a 2^{-16} cheating probability with a security of 2^{128}	82
6.1	Comparison of security requirements	93
6.2	Comparison of storage and communication performance (in bit)	96
7.1	The algorithms execution time and memory storage needed (in Bit)	116

List of Algorithms

2.1	The Stern identification protocol[123]	28
2.2	The Aguilar, Gaborit and Schrek (AGS) protocol [1]	29
2.3	Unforgeability Notions [29]	33
2.4	Anonymity notion	34
2.5	Experiment $Exp_{FDGS, \mathcal{A}}^{corr}(\lambda)$	37
2.6	Experiment $Exp_{FDGS, \mathcal{A}}^{anon-b}(\lambda)$	37
2.7	Experiment $Exp_{FDGS, \mathcal{A}}^{non-frame}(\lambda)$	38
2.8	Experiment $Exp_{FDGS, \mathcal{A}}^{trace}(\lambda)$	39
2.9	Experiment $Exp_{FDGS, \mathcal{A}}^{trace-sound}(\lambda)$	39
2.10	Collision Experiment	41
2.11	Indistinguishability Experiment	42
4.1	Setup algorithm	56
4.2	The AGS variant	58
4.3	Join interactive scheme	60
4.4	Sign ($g_{pk}, s_k[i], m, l_\lambda, \mu$)	61
4.5	Verif ($g_{pk}, m, l_\lambda, \sigma$)	62
4.6	Open algorithm	63
5.1	Key generation algorithm	74
5.2	The proposed identification protocol (RankId)	75
6.1	Our RFID Authentication Scheme	90
7.1	Function RE()	101
7.2	The Stern-Like Protocol for the relation $R_{abstract}$	103
7.3	Adversary $\mathcal{A}'$	105
7.4	Hash function h_H	106
7.5	Gen() Algorithm	107

7.6	Eval() Algorithm	108
7.7	WitCreate() Algorithm	108
7.8	TVerify() Algorithm	109
7.9	TUpdate() Algorithm	119
7.10	GSetup(λ) Algorithm	120
7.11	$\langle \text{GKgen}_{\text{GM}}(p_p), \text{GKgen}_{\text{TM}}(p_p) \rangle$ Algorithm	121
7.12	UKgen(p_p) Algorithm	121
7.13	GUpdate($g_{pk}, m_{sk}, \mathcal{I}_\tau, \mathbb{S}, \mathbf{r}$) Algorithm	122
7.14	IsActive($\mathcal{I}_\tau, \mathbf{r}, u_{id}$) Algorithm	122
7.15	Join($\mathcal{I}_\tau, g_{pk}, u_{pk}, u_{sk}$) Algorithm	123
7.16	Issue($\mathcal{I}_\tau, m_{sk}, u_{id}$) Algorithm	123
7.17	Sign($g_{pk}, gsk(j), \mathcal{I}_\tau, M$) Algorithm	124
7.18	Verify($g_{pk}, \mathcal{I}_\tau, M, \Sigma$) Algorithm	124
7.19	Trace($g_{pk}, t_{sk}, \mathcal{I}_\tau, \mathbf{r}, M, \Sigma$) Algorithm	125
7.20	Judge($g_{pk}, u_{id}, \mathcal{I}_\tau, \Pi_{trace}, M, \Sigma$) Algorithm	126

Introduction

Contents

1.1	Historical Cryptography	2
1.2	Modern Cryptography	3
1.2.1	Symmetric Cryptography	3
1.2.2	Asymmetric Cryptography	3
1.2.3	Post-Quantum Cryptography	5
1.3	Thesis Context and Related Works	7
1.4	Manuscript Organisation	14

We begin by giving a brief history of old and modern cryptography. Thereafter, we present post-quantum cryptography; espically code-based cryptogrphy. To finish, we give the related works and a presentation of our contibution.

Etymologically, cryptology comes from the Greek $\rho\upsilon\pi\tau\omicron\zeta$ that means hidden and $\lambda\omicron\gamma\iota\alpha$ that means science. Then, cryptography can be defined as the science of the secret. This science has a vital importance in communications since it is used when two peoples want to be sure that no one can read and process their exchanges. The science of secret consists in two essential disciplines that are cryptography and cryptanalysis. The cryptography has as goal to secure and protect the communications using secret codes; however cryptanalysis has as goal to break theses codes. In this manuscript we focus on the cryptography.

The cryptography uses mathematics algorithms to make an information intelligible for all excepted the recipient. The information security using the cryptography have to meet the following requirements:

- Confidentiality means the information cannot be understood by anyone for whom it is unintended;
- Integrity means the information cannot be altered in storage or transit between sender and intended receiver without the alteration being detected;

- Non-repudiation means the creator/sender of the information cannot deny at a later stage his or her intentions in the creation or transmission of the information;
- Authentication means the sender and receiver can confirm each other's identity and the origin/destination of the information.

1.1 Historical Cryptography

Throughout its history, the cryptography has get several improvements from many algorithms. Here, we present some of these algorithms. Used by the Spartan in 404 before Jesus Christ, the skytale is a first known transposition cipher scheme where the ciphertext and the plaintext have the same letters, only the orders has changed. The idea of skytale is to wrap a paper web around a staff and then write the message out in rows aligned with the staff. When the paper is taken off the staff, the order of the letters changes. To recover the message, the receiver may have a staff of the same diameter. This technique is simple but very easy to break. In fact, if someone has intercept the paper web, he can recover the staff diameter by counting the number of letter and getting all divisors of this number and then for all divisor he advances a supposition of the staff diameter. This attack is possible, because all letters of plaintext are in the ciphertext. Unlike the skytale, the Cæsar cipher is a monoalphabetic substitution cipher where a given letter is consistently replaced by the one that comes three places later in the alphabet. But another monoalphabetic substitution cipher that not just use a cyclic shift but replace arbitrary one letter in plaintext alphabet by another one. These types of monoalphabetic substitution cipher have $26!$ Possible keys that make the key guessing difficult to do. However, each letter is always enciphered in the same manner that make all monoalphabetic substitution cipher easy to break using a frequency analysis. Because of the frequency analysis attack, we saw above that the monoalphabetic cipher is not secure. To combat this attack, the polyalphabetic substitution cipher which encode a character in a text in deferent way is a good candidate. A famous example of polyalphabetic cipher called Vigenère cipher. Enigma was the famous cryptographic machine used by the German during the Word War II. Enigma uses both permutation and transposition operations that make it a very complex cryptographic system. Depending on its version, Enigma is composed by at most five rotors that act like a big gearing, so that a complete round of a first rotor imply one move of a second rotor and so on. Besides the rotors, Enigma add to its composition a table of permutation, a mirror and a keyboard. A tape on the keyboard produces one move on the first rotor that shuffle the initial configuration. The way to decrypt one message is to find the initial configuration. One of the famous contribution

of cryptography to stop the Word War II is the break of Enigma by Bletchley Park and Alan Turing. It is believed that the break of Enigma has shortened the Word War II by two years.

1.2 Modern Cryptography

The coming of the computer gave a new skin of the cryptography which was only for military and diplomatic uses. The simple fact that it is possible to perform hundreds operations in a second is an opportunity to design a more complex and secure cryptosystems. There are two kind of cryptography: the symetric cryptography and the asymetric cryptography.

1.2.1 Symmetric Cryptography

Symmetric Cryptography uses a same key to encrypt and to decrypt a message. Symmetric cryptosystem is devised in two big categories. The first category is the stream cipher that crypt the message by buffer regarding the size. And the second category is the bloc cypher, here the message is split in blocs of same size and crypt each bloc by using permutation and substitution. The most popular bloc cypher was the Fresnel scheme [60] and has been standardized in 1977 as Data Encryption Standard (DES) algorithm. Later, du to its smaller key size, using the recent computers, it is possible to recover the DES secret key in a few time. Then to get a bigger key size, Triple-DES [117] has been proposed in 1996. Parallely, a contest has been launched by the National Institute of Standards and Technology (NIST) for the Advanced Encryption Standard (AES) to replace DES. From this contest, it was output that the proposed of Johan Damien and Vincent Rijmen [51] became the AES in 2001. For now, AES is still secure and is widely used by many applications and processors. Symmetric cryptography is very speed but need that booth the sender and the receiver know the secret key before the encryption and decryption operations. To avoid this problem, Whitfield Diffie and Martin Hellman proposed in [55] the use of two different keys: one for the encryption which will be public and another one which remains private for the decryption. In [55] they also proposed a method to get a common secret remotely. This kind of scheme is called asymmetric cryptosystem.

1.2.2 Asymmetric Cryptography

The most disavantage of of symetric cryptography is the sharing of the key. Unlike symetric cryptography, Asymmetric Cryptography, called also public key cryptography, uses two differ-

ent keys: one key, called public key, to encrypt the message and another key, called private key, to decrypt the cypher-text. The first asymmetric Cryptography was proposed in 1978 by Ronald Rivest, Adi Shamir and Leonard Adleman [115]. The Rivest Shamir Adleman (RSA) cryptosystem is widely used in bank applications. In RSA we choose two big prime numbers p and q , and we compute $N = p \cdot q$. Thereafter we choose an integer e prime with $\Phi(N) = (p-1) \cdot (q-1)$ and we compute the integer d such that $e \times d = 1 \pmod{\Phi(N)}$. The public key is the couple (N, e) and the private key is d . Here to encrypt a message $m \in [0, \dots, N[$ we compute $C = m^e \pmod{N}$ and to decrypt a message we compute $C^d = (m^e)^d = m^{e \cdot d} = m^1 = m \pmod{N}$. Thereafter, the concept of using two different keys allow the emergence of signature scheme and Zero-Knowledge identification (See Chapter 2). In these type of scheme the private key is used to sign the message and anyone can use the public key to verify the signature.

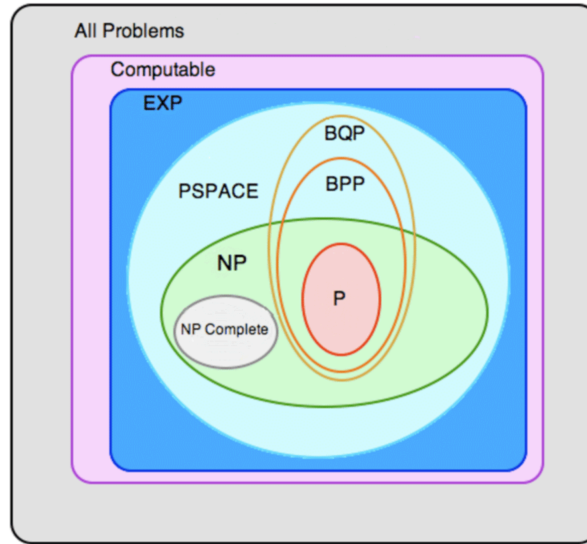


Fig. 1.1: Venn Diagram of Complexity Classes

The main tool to evaluate the strength of a cryptographic scheme is the complexity. The complexity of a problem is the number of operations needed to solve this problem. As it is shown in Figure 1.1, complexity is organised in many classes. In cryptography only the class P which resume the decisional problems that there exist a determinist solution that can be evaluate in polynomial time. Also, the NP class that resume the decidable problems that we can not find a solution in polynomial time. The most public key cryptographic scheme have there security based on NP class problems, it is the case of RSA or the Diffie-Hellman cryptosystem.

Remark 1. Figure 1.1 is commented as follow:

P : Solution can be found in polynomial time in the data size.

NP : Solution can be checked in polynomial time.

NP COMPLETE: Any NP problem can be reduced to one of these.

BPP: Solution in polynomial time at probability $> 1/2$.

BQP: Solution in polynomial time at probability $> 1/2$ on quantum computer.

PSPACE: Solution requires a polynomial amount of memory.

EXP: Solution can be found in exponential time.

COMPUTABLE: Solution can be found eventually.

The RSA problem is to find a e^{th} root modulo the number N , then it is the same thing to find the factorisation of N which is $p \cdot q$ to after compute $\Phi(N) = (p - 1) \cdot (q - 1)$ and compute the inverse of $e \bmod \Phi(N)$ to finally find the secret d . Then it is clear that the RSA problem is reduced to the factorisation problem. For now, the best know algorithm to solve the factorisation problem is the algebraic sieve [113] which have a complexity of $L_n \left[\frac{1}{3}, \sqrt[3]{\frac{64}{9}} \right]$ where the complexity function L_n is as follow:

$$L_n[\alpha, c] = \exp^{(c+o(1))(\ln n)^\alpha (\ln \ln n)^{1-\alpha}}, \text{ where } 0 < \alpha < 1. \quad (1.1)$$

The Diffie-Hellman problem is also based on the discrete logarithm problem. Concretely, the discrete logarithm problem consist in inverting the exponential function in a group $(\mathbb{F}_q^*, \times)$. Given a prime number p and $g \in \mathbb{F}_p^*$ a generator of $\mathbb{F}_q^*$, the Diffie-Hellman protocol allow two challengers to get a common secret. At this moment, there is no yet a generic algorithm to solve the Diffie-Hellman problem. Du to there efficiency, the discrete logarithm and the factorisation problems are the most used problem in the modern cryptography like in signature scheme, e-voting schemes, e-bidding, digital right management systems, controlled anonymous printing services and other domains.

1.2.3 Post-Quantum Cryptography

In 1994, assuming the existence of a quantum computer, Shor [121] showed that it is possible to solve the factorisation problem in $\mathcal{O}((\log n)^2 (\log \log n) (\log \log \log n))$ time. In other words, the discrete logarithm and the factorisation problems become obsolete.

The quantum computer is based on two principles: The fist one is the use of q-bits. Instead of classical bit that can take a value 1 or 0, the q-bit which can be 1, 0 or the superposition of these two state. The second principle is the quantum entanglement. This phenomenon occur between two particles when they are make together a linked system. From these two principles, an entangled q-bit vector of size n can take simultaneously all of 2^n possible states of classical

bits vector. But at the end of the computing, the q-bit vector take only one of the 2^n states (the most probable). It is clear that the existence of a such algorithm allow the emergence of alternatives cryptosystem such as the post-quantum cryptography.

In 2016, the NIST has lunch a contest to develop a new cryptographic algorithms that would be secure against both quantum and classical computers. These algorithms could serve as replacements for our current public-key cryptosystems to prepare for the eventuality that large-scale quantum computers become a reality. The most known post-quantum cryptography candidates such as Lattice-based cryptography, Isogeny-based cryptography, code-based cryptography and multivariate-based cryptography have their security linked on difficult problems for which their exist no classical or quantum algorithms to solve them.

Lattice-based Cryptography Introduced in 1996 by Miklós Ajtai [2], lattice-based cryptography is based on the hardness of the well studied lattice problem called Short Integer Solutions (SIS). The SIS problem consists in finding a non trivial solution to a linear equation modulo an integer, so that the coefficients of this solution are not too big. Two years latter, Jeffrey Hoffstein, Jill Pipher and Joseph H. Silverman introduced a famous lattice-based public-key encryption scheme called NTRU [80]. But the NTRU cryptosystem has not been proved at least as hard as solving a worst-case SIS problem. In 2005, using the Learning with Errors (LWE) problem, Oded Regev [114] introduced the first lattice-based public-key encryption scheme whose security was proven under worst-case hardness assumptions. Since then, much more work has been devoted to constructing another cryptographic primitive based on SIS and LWE. The famous example of cryptographic primitives introduced from lattice-based cryptography is the fully homomorphic encryption [71] by C. Gentry.

Isogeny-based Cryptography In 1997, Couveignes introduces the Hard Homogeneous Spaces framework [49] which is the fundamental of isogeny-based cryptography. But his work stayed unpublished for 10 years. In 2006, Rostovtsev and Stolbunov [116] independently rediscover Couveignes ideas, suggest isogeny-based Diffie-Hellman as a quantum-resistant primitive. Since then, many others isogeny-based cryptosystems have been developed. One of them, is an efficient post-quantum key exchange called SIKE and introduced by Jao and Feo [84]. SIKE is the only isogeny-based candidate submitted to the NIST competition.

Multivariate-based Cryptography A Multivariate-based Cryptography or a Multivariate Public-Key Cryptosystem (MPKC) is a public-key cryptosystem where the public map, or trapdoor one-way function, is given as a set of m polynomials of a small degree d equations over n

variables in a finite field where m , d and n are non null integers. Even if the idea of using Multivariate polynomial in cryptography appear early [56, 61], it is in 1988 that T. Matsumoto and H. Imai [102] presented the first MPKC in the modern form. Since this paper, many other work for MPKC has been proposed.

Code-based Cryptography In 1949, to establish a communication without noise, Shannon [119] developed the theory of error correcting code. The main idea of Shannon is to use error correcting codes to detect and correct errors after the reception of a message from noisy channel. Here, each message m is represented by an n -tuple $(m_1, \dots, m_n)$ called word of length n also the elements $(m_i)_{i=1, \dots, n}$ are from a set $\mathbb{A}$ called the alphabet and the set of all words obtained is called a code of length n over $\mathbb{A}$. We have to wait 1978, to see the first code-based cryptosystem proposed by Robert McEliece [103]. The most important code characteristic is the quantity of error that a code can detect and correct. To measure these quantity we need to use metrics. In error correcting codes we commonly use two metrics: the Hamming metric and the rank metric.

In this thesis we focus on code-based cryptography. Code-based cryptography uses two different metrics: the first metric is the Hamming metric which is very known and studied and the rank metric which is less known, but have a good promises. These two metrics propose a difficult problems which are very useful in many primitives cryptographic.

1.3 Thesis Context and Related Works

The goal of this thesis is to contribute to code-based cryptography which is supposed to be resistant to the quantum computer attacks. Our contribution focus on rich cryptographic algorithms like signature scheme, authentication scheme for Radio Frequency Identification (RFID), Zero-Knowledge identification scheme and cryptographic. Hereafter we describe our contributions.

A Code-based Group Signature Scheme with Shorter Public Key Length [8] Group signatures allow members to sign on behalf of a group while maintaining signers identity anonymous. Group signature was introduced in [42]. Since then, many works have been proposed in this area [11, 30, 37, 75]. For anonymity and traceability properties of these schemes, group signatures are highly useful in various real-life scenarios such as controlled anonymous printing services, digital right management systems, e-bidding and e-voting schemes. The majority

of group signature schemes are based on number theory assumptions [30, 32, 38, 94], but number-theoretic based cryptography will not resist to the quantum computing. Recently, the research for post-quantum group signatures is quite active as shows these publications [3, 75, 89, 90, 95]. The majority of these works are based on lattice assumptions, while in code-based cryptography we denote two group signature schemes: the first one is based on BSZ model [22] and it is presented in [3]. The second one is based on BMW model [20] and it is presented in [58]. It is known that one of the big issue in code-based cryptography is the size of the public key. While in [58], Ezerman *et al.* proposed a first provably secure code based cryptography regardless on the size of the public key. Because of the big size of the public key, it is not practical to use the group signature scheme. In the Chapter 3 of this thesis, we show that it is possible to reduce the public key length of the first provably secure group signature scheme from code-based assumptions without losing the security properties. More precisely, the public key can be 466 times shorter than the original scheme, typically for a group of 16 users when the public key length is 1,34 kilo-bytes, while the size is 625 kilo-bytes in the original scheme [58]. Our technique consists in using a QC-MDPC variant for encrypting user identity and a random double circulant matrix for the underlying Zero Knowledge Argument System.

A New Dynamic Code-Based Group Signature Scheme [14] Group signature is a cryptographic primitive where a user can anonymously sign a message on behalf of group users. The dynamic case in group signature is more interesting than the static one. For many years, several group signatures have been introduced namely the ACJT [11], which was the first provably secure coalition-resistant scheme under the Strong RSA and Decisional Diffie-Hellman (DDH) assumptions. Boneh, Boyen and Shacham in [30] and later Camenisch and Lysyanskaya in [36] proposed very efficient group signature schemes using bilinear maps. Independently, Nguyen and Safavi-Naini presented in [108] another group signature scheme using bilinear maps. We note that all these schemes are analyzed in the random oracle model [21]. Bellare *et al.* in [20] gave formal definitions of the security properties of group signatures, they proposed the first scheme provably secure in the standard model (while totally unpractical). Independently, Kiayias and Yung [86] (and later [87]) defined a security model. Bellare *et al.* [22] extended the [20] model to the case of dynamic groups. All the aforementioned group signature schemes are based on number theory assumptions. However, number-theoretic based cryptography will not resist to the quantum computer. Recently, the research for post-quantum group signatures is quite active as shows these publications [3, 8, 75, 89, 90, 95]. The majority of these works are based on lattice assumptions, while in code-based cryptography we denote three group signature schemes. The two first schemes presented in Asiacrypt 2015 [58] and [8] in Secrypt 2016 are based on BMW (Bellare, Micciancio and Warinschi) model [20]. The last one is based on a slight relaxation of the BSZ (Bellare, Shi and Zang) model [22] and is proposed in [3]. In [3],

Quentin *et al.* proposed the first dynamic code-based group signature. But all proposed scheme uses trapdoor matrix which is actually hard to find and the only current candidate for instantiating there construction is the Courtois-Finiasz-Sendrier (CFS) matrix which is not efficient in practice. In the Chapter 4 of this thesis, we propose a new dynamic code-based group signature without trapdoor matrix. The general idea of this scheme consists in finding a collision between two ciphertexts using two different QC-MDPC matrices in McEliece cryptosystem. We use a variation of AGS Zero-Knowledge protocol to prove the possession of the secret key and then we use the Fiat-Shamir transformation to turn it into a signature. The public key and signature sizes are constants and independent of group users size and are shorter than those presented in the literature for 80 bits security level. Furthermore, the proposed group signature scheme presents several advantages: it is a dynamic group signature based on error correcting code assumptions which are supposed to be resistant to quantum computing.

An Efficient Identification Scheme Based on Rank Metric [15] Identification protocols are very useful in the electronic communications area. For example, identification protocols are used in ticketing, e-passport, e-health, control access systems, etc. The choice of code-based cryptography for identification schemes is mainly motivated by the capacity of future quantum computer. An identification scheme is said to be Zero-knowledge if it is an interactive protocol where one proves the knowledge of something, without revealing any information on this knowledge. This implies that the verifier cannot learn the secret, also he or she cannot replay the protocol and especially he or she cannot convince another person that the prover was aware of the secret. Over the years, many Zero-knowledge identification schemes have been proposed using Hamming metric [1, 40, 123, 127]. Based on the instances of Syndrome Decoding problem, Stern presented in [123] a pioneering work that is the first code-based Zero-knowledge identification scheme whose cheating probability is $\frac{2}{3}$. In this scheme, the prover has to convince the verifier that he or she knows a word e of small weight such that $He^\top = y$ where the matrix $H \in \mathcal{M}_{(n-k) \times n}(\mathbb{F}_2)$ which is a parity check matrix of a linear binary code of length n and dimension k and the vector $y \in \mathbb{F}_2^{(n-k)}$ are public parameters. Thereafter, to decrease the cheating probability to $\frac{1}{2}$, Stern proposed in [123] an extended version of his first protocol that consists on five pass rather than three pass. This improvement impacts a reduction of the signature size. But on the other hand, it increases the communication cost. The second advanced work was presented in [127] by Veron where he proposed a different formulation of secret key using the General Decoding (GD) problem (see below). In this case the prover has to prove his knowledge of a couple of vectors (f, e) that verifies $x = fG \oplus e \in \mathbb{F}_2^{(n-k)}$ where the matrix $G \in \mathcal{M}_{n \times k}(\mathbb{F}_2)$ and x are public parameters. This new formulation decreases the communication cost but increases the key size. In [65], Gaborit and Girault proposed to use a double circulant matrices to represent the public matrix. This proposition enables to decrease

dramatically the public key size. In [40] using a q -ary codes, Cayrel *et al.* proposed another protocol with cheating probability of $\frac{1}{2}$. The last and the most efficient protocol in Hamming metric has been presented by Aguilar *et al.* in [1] which is based on the Veron's scheme and achieves a cheating probability of $\frac{1}{2}$ while preserving a low communications cost. Chen in [44] proposed a new code-based Zero-knowledge identification scheme using another metric called rank metric. This new three pass identification scheme differs from Stern protocol [123] on the following points:

- The lack of hash function; the first version of Chen identification scheme [44] did not use hash function but matrix multiplication. This property has been very useful since the use of hash function increases the complexity of the identification scheme.
- The choice of another family of error correcting codes; Stern has used a family of error correcting codes using Hamming metric but Chen protocol [44] has used codes in rank metric. This choice can provide smaller key size than Hamming metric.

A few month after the publication of Chen's identification scheme [44], Chabaud and Stern have presented in [41] the first attack on the syndrome decoding problem in rank metric. Thus, to keep Chen's protocol safe, Chabaud and Stern [41] proposed to recalibrate the security parameters. However, the second attack presented by Gaborit *et al.* in [67] that is based on the lack of a hash function allows a passive attacker to quickly recover the user secret. To repair the Chen's protocol, Gaborit *et al.* [67] proposed a new version with hash function. This new protocol can be seen as the rank version of Stern's protocol. Recently the research in rank metric codes applied to the cryptography area is quite active with the publication of many paper like [67, 79]. In the Chapter 5, we present a five pass Zero-knowledge code-based identification scheme in metric rank with a cheating probability of $\frac{1}{2}$. Like in AGS protocol [1], our protocol uses double-circulant matrices to represent public generator matrix. Moreover the matrices used to mask user secrets in the protocol running are also circulant. This technique combined with rank metric instead of Hamming metric provide an efficient identification scheme with small key size and low communications cost. In practice, our scheme is more efficient since the public data are 276 bits rather than 350 bits in AGS protocol for 80 bits security level which implies a reduction of 21%. Moreover, for a security level of 128 bits, we get 434 bits as public data size that is better than 550 bits in the AGS protocol, which implies a gain of 21%. Also, for a security of 2^{80} , we get a signature size of 83160 bits which is better than AGS [1] case where the signature size is about 93000 bits. By applying the Fiat-Shamir paradigm [62] to a Zero-knowledge identification scheme it is easy to obtain a signature scheme.

Rank-Metric Code-Based Mutual Authentication Protocol for RFID [16] Thanks to their small size, low cost, low voltage and long lifetime, Radio Frequency Identification (RFID) systems are becoming increasingly popular and are used more and more frequently in applications such as access control, supply chain management, e-health and so on. RFID is an automatic identification technology that uses radio frequency radiation to identify tagged objects. We consider RFID system with Reader and Tag only, described as follows:

Tag, denoted as T , it is a small electronic chip supplemented with an antenna that can transmit and receive data. RFID Tag can be divided into three categories: passive Tag, which is powered by the radio wave from an RFID Reader and communicates with the Reader through backscattering. Active and semi-passive RFID Tags use internal batteries to power their circuits. An active Tag also uses its battery to broadcast radio waves to the Reader, whereas a semi-passive Tag relies on the Reader to supply its power for broadcasting.

Reader, denoted as R , it is a device to communicate with Tag by radio waves.

The authentication process of RFID Tags by an RFID Reader is necessary to ensure the authenticity, the freshness and the presence of the RFID Tags. During this authentication process, RFID Tags can check the authenticity, the freshness and the presence of the Reader, which is so-called mutual authentication. The main issues in RFID systems are the constraint of resources and the security requirements. In addition to anonymity and untraceability, an authentication protocol for RFID systems must verify the following six properties as defined in [47] namely: Mutual Authentication, Desynchronization Resilience, Anonymity, Untraceability, Forward Secrecy and Resistance to Replay Attack. Our protocol also verifies the Full Inside Secrecy property, which is an extension of the Inside Secrecy property presented in [10]. We define the Full Inside Secrecy property as follow:

Full Inside Secrecy means that even if we have a genuine corrupted Reader with some secrets, an adversary cannot extract the private keys from the Tags. Furthermore if an adversary can corrupt one or many Tags, he or she cannot produce a Reader which can be authenticated by a Tag which has not been corrupted.

To figure out the importance of this property, consider a practical example of a shop where each article is equipped with a Tag. In this case, no one should be able to pick up an article and get out from the shop without paying it. But in various authentication schemes, if someone can get access to a Tag memory then he or she can get the necessary information to produce a

reader which can be accepted by another Tag and then get the article marked as paid. The Full Inside Secrecy property gives a direct response to the previous issue by ensuring that even if an adversary can corrupt one or many Tags, he or she cannot produce a Reader which can be authenticated by a Tag which has not been corrupted. The Full Inside Secrecy property also ensures the Inside Secrecy property, which means that even if we have a vulnerable Reader, an adversary cannot obtain any more Tag's identifier since this identifier has been never revealed by the Tag in the authentication process. In the literature there exist four types of RFID authentication protocols. The first one consists of protocols that support cryptographic functions whether symmetric or asymmetric [101]. The second type is composed of protocols that use pseudo random generator number PRNG and hash functions [10, 52, 82]. The third type of protocols called lightweight protocols is generally based on some error correcting codes such as Cyclic Redundancy Check (CRC) and checksum [17, 106, 129]. The last type of protocols called ultra-lightweight are those protocols that use only basic binary operations like XOR, AND and NOT [81, 122]. Over the years, many authentication protocols for RFID systems have been proposed and the majority of them are based on number theory and elliptic-curve cryptography [43, 106]. Hung-Yu [82] presented in 2006 an RFID authentication protocol which provides the anonymity property besides other properties. Later, Naved *et al.* [106] presented a reliable and low cost RFID authentication protocol. In [50] Cui *et al.* proposed a code-based authentication which does not achieve forward secrecy property. Thereafter, in [46] Chien and Laih used a confusion scheme to ensure the untraceability of their protocol based on error correcting codes. However, it does not achieve the inside secrecy property. In [100] Malek *et al.* based their authentication protocol on randomized McEliece cryptosystem, however this scheme cannot achieve desynchronization resilience. In [92], a mutual authentication code-based protocol is presented by Li *et al.* However, this scheme is not resistant to traceability and forward secrecy attacks. A year later, this scheme was improved by Chikouche *et al.* [47] to achieve untraceability and forward secrecy properties. In [10], two mutual authentication protocols using code-based Zero-Knowledge identification scheme proposed by Assidi *et al.* [10], that achieved Inside Secrecy. However an inconvenience of [10] is the fact that it is possible to produce a reader which can be authenticated by a tag which is not corrupted. With the hypothetical deployment of quantum computers, code-based cryptography is a valid alternative to number theoretical cryptography. Besides, code-based cryptography is used in many cryptographic primitives such as identity-based cryptography, group signature and ring signature [9, 14, 66, 104]. In the Chapter 6, we present the first mutual authentication protocol for RFID systems based on rank metric codes which does not use Zero-Knowledge identification schemes as in [10]. Furthermore, our protocol uses two improvements, the first one consists in using double circulant low rank parity check codes. This improvement permits to reduce communication costs and to get smaller key size than previous code-based protocols. The sec-

ond security enhancement consists in using a hash function to get full inside secrecy property and the following properties: mutual authentication, desynchronization resilience, anonymity, untraceability, forward secrecy and resistance to replay attacks.

A Code Based Cryptographic Accumulator And Fully Dynamic Group Signature The cryptographic accumulator notion was first introduced by Benaloh and De Mare [24] in 1993. A cryptographic accumulator is a cryptographic primitive which allows accumulating a finite set of elements into a single value that is called *Accumulated Value*. Therefore, for each element, it is possible to generate a short membership proof called Witness, which attests the belonging of the element to the set. The accumulated value is public. For a cryptographic accumulator to be secure, it should not be possible to find a valid witness for an element not belonging to the set. In the classical example, to accumulate a finite set $X = \{x_i\}$ of big prime integers, we compute the product x of all elements of X . To prove the membership of an element x_i of X , we just compute $q_i = \frac{x}{x_i}$ where q_i is an integer and is called the witness of x_i . Then everyone can check the membership of x_i by checking if $x_i q_i = x$ which is an easy computation. But the factorization of x to extract x_i is hard for usual computers but not for the hypothetical quantum computer. In the first scheme introduced by Benaloh *et al.* [24], it is not possible to add or remove elements from the set of accumulated values. This type of accumulator is said to be static. Thereafter to overcome this lack, in 2002 Camenisch and Lysyanskaya [35] proposed a first dynamic accumulator based on RSA assumptions which allow elements to be added or removed from the set of accumulated values, but they do not have an efficient non-membership proof. Later, many other constructions have been proposed like the ones based on bilinear pairing [34] and vector commitments [39]. But all of them cannot allow providing non-membership proofs. Li *et al.* proposed in [91] a construction of an accumulator which supports both membership and non-membership proofs. But the proposed scheme requires a trusted accumulator manager and does not completely fit the accumulator definition given firstly by Benaloh *et al.* namely, to provide a membership proof without necessarily the need of a trusted accumulator manager. To fix this lack, Camacho *et al.* [33] proposed a strong universal accumulator which does not need a trusted accumulator manager. All of the schemes proposed above have their security based on the discrete logarithm or the factorization problems that are not resistant to the quantum computer. This fact among others has enabled the emergency of post-quantum cryptography, like lattice-based cryptography, code-based cryptography, multivariate cryptography or isogeny-based cryptography. Then in 2015, based on lattice assumptions, a first direct construction of post-quantum cryptographic accumulator has been proposed by Jhanwar and Safavi-Naini in [85]. One year later, using Merkle-Tree, Libert *et al.* [93] presented another lattice-based cryptographic accumulator. Recently using the same technique, Derler *et al.* [54] presented a post-quantum accumulator from symmetric-key

primitives, but their scheme is also static and cannot support non-membership proofs. In 2018, Ling *et al.* [96] proposed a dynamic lattice-based accumulator scheme by adding to Libert *et al.* [93] scheme an efficient update algorithm. Also in 2018, Yu *et al.* [128] proposed a lattice-based universal accumulator by adding the support of non-membership proofs to Libert *et al.* [93] scheme. In 2019, using Merkle-Tree, Nguyen *et al.* [107] proposed a first code-based cryptographic accumulator. Their proposed scheme uses random codes which need a big storage capacity and they do not provide an implementation of their scheme. Cryptographic accumulators are very useful in many privacy-preserving technologies like anonymous credential systems, group signatures, efficient time-stamping, sanitizable signatures, ring signatures, redactable signatures, identity-based encryption, Zerocoin, *etc...* (for more detail see [53]). Group signature is a cryptographic primitive where a user can anonymously sign a message on behalf of group users. In dynamic group signature, the group manager has the possibility to add or remove users. When a group signature scheme allows to add and remove user it is called fully dynamic group signature (*FDGS*) scheme. The first post-quantum *FDGS* scheme is from lattice-based cryptography [96]. In code-based cryptography, as far we know, they are two dynamic group signature schemes [3, 14] which allow the group manager to just add users. In Chapter 7, we propose a new code-based cryptographic accumulator. Like in [93], our cryptographic accumulator proceeds by computing a Merkle tree using a collision-resistant family of hash functions based on a $[2k, k]$ -code over the binary field $\mathbb{F}_2$. Moreover, this technique uses double circulant codes that provide compact keys and is securely based on the hardness of the syndrome decoding problem. Furthermore, to support the feasibility of our scheme, we give an implementation of our cryptographic accumulator which is also, as far as we know, the first direct implementation of post-quantum cryptographic accumulators. Indeed, the others are the implementations of rich cryptographic primitives that use cryptographic accumulators. Our construction allows to deduce a fully dynamic code-based group signature scheme which is logarithmic complexity on the group size.

1.4 Manuscript Organisation

The remainder of this thesis is organized as follows. Chapter 2 is devoted to definitions and backgrounds. In Chapter 3, we propose a technique to decrease the public key size of the signature scheme presented in [58]. After, we propose in Chapter 4 a new dynamic code-based group signature scheme. Thereafter in chapter 5, we design an efficient identification scheme based on rank metric. Chapter 6 is devoted to our new rank-metric code-based mutual authentication protocol for RFID and in Chapter 7 we proposed a code based cryptographic

accumulator and a fully dynamic group signature scheme. Finally, we make a conclusion with some open researches axis.

Definitions and Background

Contents

2.1	Linear Codes	16
2.1.1	Hamming metric	17
	The General Decoding Problem	18
	Random Double-Circulant Codes	19
	The McEliece Cryptosystem	20
	QC-MDPC Codes	21
2.1.2	Rank metric	23
	Properties of the Asterisk Product	25
	The General Decoding Problem in Rank Metric	25
	The McEliece Cryptosystem in Rank Metric	26
2.2	Zero-Knowledge Identification Scheme	27
2.3	Group Signature	29
2.3.1	Static Group Signature	29
2.3.2	Dynamic Group Signature	31
2.3.3	Fully Dynamic Group Signature	34
2.4	Cryptographic Accumulator	38
2.5	Conclusion	42

To make this thesis self contained, in this chapter, we give some definitions on difficult problems behind coding theory from Hamming metric and the rank metric codes. Second, we recall fundamentals of zero-knowledge identification scheme. Third, we recall definition and security model for group signature. And we finish this chapter with the definition and security model for cryptographic accumulators.

2.1 Linear Codes

In 1950, Hamming introduced the linear codes[78] by designing the Hamming code which encodes four data bits into seven bits by adding three parity bits. Informally, a linear code is an

error-correcting code for which any linear combination of codewords is also a codeword. We can give a formal definition of a linear code.

Definition 1 (linear code). *An (n, k) -linear code of length n and dimension k over the finite field $\mathbb{F}_q$ is a subspace of $\mathbb{F}_q^n$ of dimension k .*

In order, an important code characteristic is the quantity of error that a code can detect and correct. To measure these quantity we need to use a metric. In error correcting codes we commonly use two metrics: the Hamming metric and the rank metric.

2.1.1 Hamming metric

Let $\mathbb{A}$ be a finite alphabet, n a non null integer and d the function as follow:

$$\begin{aligned} d &: \mathbb{A}^n \times \mathbb{A}^n \rightarrow \mathbb{N} \\ (x, y) &\mapsto d(x, y) = \#\{i \in \{1, \dots, n\} : x_i \neq y_i\} \end{aligned} \quad (2.1)$$

where $\#\mathbb{S}$ means the cardinal of the set $\mathbb{S}$.

Proposition 1. *The function d is a metric on $\mathbb{A}^n$.*

We define from the Hamming distance (metric) the weight of a vector of $\mathbb{A}^n$.

Definition 2 (Hamming weight). *The Hamming weight of a vector $v \in \mathbb{A}^n$ denoted $wt(v)$ is the distance between the vector v and the vector null of $\mathbb{A}^n$. Then we can write $wt(v) = d(v, 0_{\mathbb{A}^n})$*

Now we can easily define the minimum distance of a code.

Definition 3. *Let $\mathcal{C}$ be a code over $\mathbb{F}_q$. The minimum distance d of $\mathcal{C}$ is the integer*

$$d = \min\{d(x, y) : x, y \in \mathcal{C}, x \neq y\} \quad (2.2)$$

From the minimum distance we define the packing radius or the correction capacity by:

$$t = \lfloor \frac{d-1}{2} \rfloor \quad (2.3)$$

Another important notion in error correcting code is the decoding. When a word c is transmitted over a noisy channel, the received word is $y = c + e$ where e is the error inducted by the noisy

channel. Then the receiver must to recover the original codeword this operation is called the decoding.

Proposition 2 (Unique decoding). *Let $\mathcal{C}$ be a code of length n over $\mathbb{F}_q$ with a minimum distance d and correction capacity t . For $y \in \mathbb{F}_q^n$, there exists at most one codeword $c \in \mathcal{C}$ such that $d(y, c) \leq t$.*

Proof. Let $y \in \mathbb{F}_q^n$. We assume that there exist $c_1, c_2 \in \mathcal{C}$ such that $d(y, c_1) \leq t$ and $d(y, c_2) \leq t$. We have

$$d(y, c_1) + d(y, c_2) \leq 2t \leq d - 1 \quad (2.4)$$

and thus,

$$d(c_1, c_2) \leq d - 1 \quad (2.5)$$

Since d is the minimum distance it is clear that $c_1 = c_2$. □

Due to their linear properties, linear codes are very useful in code-based cryptography. We give in the following some definitions of linear codes. Lets n, k and q be non null integers and let $\mathbb{F}_q$ be a finite field of q elements. Usually, we use (n, k) -code to denote an (n, k) -linear code. Also we will use (n, k, d) -code to denote an (n, k) -linear code of minimum distance d . An (n, k) -code can be represented by a specifics matrices: the generator and the parity-check matrices.

Definition 4 (Generator Matrix). *Let $\mathcal{C}$ be an (n, k) -linear code over $\mathbb{F}_q$. We call a generator matrix of $\mathcal{C}$ a $k \times n$ matrix G whose rows form a basis of $\mathcal{C}$. We can then write:*

$$\mathcal{C} = \{m \cdot G, m \in \mathbb{F}_q^k\} \quad (2.6)$$

Definition 5 (Parity-Check Matrix). *Let $\mathcal{C}$ be an (n, k) -linear code over $\mathbb{F}_q$. We call a parity-check matrix of $\mathcal{C}$ a matrix H such that for all word c of $\mathcal{C}$ we have $H \cdot c^\top = 0$*

The General Decoding Problem

In the following, we define the general decoding problem which is used by many code-based cryptosystem. The general decoding problem is the problem of decoding a received word to the closest codeword in an arbitrary code. It was first used in 1978 by McEliece[103] to prove the security of his cryptosystem. The Syndrome Decoding (SD) problem (Problem 2) and the GD problem (Problem 1) are two problems based on coding theory proved to be NP-complete in [26].

Problem 1 (GD Problem[26]). The $GD(n, n-k, \omega)$ problem is defined as follows: Let k, n and w be integers, and let (G, x, ω) be a triple consisting of a matrix $G \in \mathcal{M}_{n-k \times n}(\mathbb{F}_2)$, vector $x \in \mathbb{F}_2^n$ and an integer $\omega < n$. Find a vector $m \in \mathbb{F}_2^{n-k}$ and a vector $e \in \mathbb{F}_2^n$ such that $wt(e) \leq \omega$ and $x = m \cdot G + e$.

General decoding problem can be formulated using the parity-check matrix, it is called syndrome decoding problem.

Problem 2 (SD Problem[26]). The $SD(n, k, \omega)$ problem is formulated as follows: Let n, k and ω be integers, given a uniformly random matrix $H \in \mathcal{M}_{k \times n}(\mathbb{F}_2)$ and a uniformly random syndrome $y \in \mathbb{F}_2^k$, find a vector $s \in \mathbb{F}_2^n$ such that $wt(s) \leq \omega$ and $H \cdot s^\top = y^\top$.

Random Double-Circulant Codes

We say that a matrix H is double circulant if $H = [B|A]$ where B and A are circulant matrices of length k , i.e. a $k \times k$ matrix generated from its first row $a = (a_0, \dots, a_{k-1})$

$$A = \begin{pmatrix} a_0 & a_1 & \cdots & a_{k-1} \\ a_{k-1} & a_0 & \cdots & a_{k-2} \\ & & \ddots & \\ a_1 & a_2 & \cdots & a_0 \end{pmatrix} \quad (2.7)$$

We define a new problem, which adapts the syndrome decoding problem to the case of random double circulant codes:

Definition 6. The $SD(2p, p, \omega)$ problem is formulated as follows:

Instance: given a random double circulant matrix $\mathbf{H} \in \mathbb{F}_2^{p \times 2p}$ and a vector $\mathbf{y} \in \mathbb{F}_2^p$.

Question: find a vector $\mathbf{s} \in B(2p, \omega)$ such that $\mathbf{H} \cdot \mathbf{s}^\top = \mathbf{y}^\top$.

It is not known whether this problem is NP-complete, but the problem is probably as hard as syndrome decoding problem, and on practical point of view (see [65]) the practical security is almost the same for the best known attack that syndrome decoding for random matrices.

The McEliece Cryptosystem

Let $G \in \mathcal{M}_{k \times n}(\mathbb{F}_q)$ be a generator matrix of an (n, k) code over $\mathbb{F}_q$ which can correct t errors in polynomial time. The McEliece cryptosystem is described as follow: Keys generation is achieved by the following steps:

1. Randomly choose $n \times n$ permutation matrix P and a $k \times k$ invertible matrix S over $\mathbb{F}_q$.
2. Compute $G' = S^{-1} \cdot G \cdot P^{-1}$.
3. Output the public key (G', t) and the private key (S, G, P) .

To encrypt a vector $m \in \mathbb{F}_q^k$ we use the following steps:

1. Choose a random vector $e \in \mathbb{F}_q^n$ of Hamming weight $\leq t$.
2. Compute the cipher text $c = m \cdot G' + e$.

To decrypt a cypher text $c \in \mathbb{F}_q^n$ we use the flowing steps:

1. Compute $y' = c \cdot P$.
2. Use an efficient decoding algorithm to find the vector $y = mS^{-1}$.
3. Compute $S \cdot y$ to get the plaintext m .

The security of McEliece cryptosystem depend on the family of codes used, and the chosen parameters. In his original paper, McEliece [103] proposed to use the $(1024, 524, 101)$ -binary Goppa code. Then it is clear that its security is based on two difficult problems:

- The difficulty of decoding a random linear code.
- The difficulty of recovering a decoding algorithm from a public matrix representation of a binary Goppa code.

In table 2.1, we give some parameters of the McEliece cryptosystem as described in [103].

Table 2.1: Parameters and key size for McEliece with Goppa codes [28]

Security level (bits)	(n, k)	t	Public Key size (bits)
80	(1632, 1269)	33	460647
128	(2960, 2288)	56	1537536
256	(6624, 5129)	115	7667855

Due to their big size, these parameters cannot be used in practice. There is a way to reduce considerably the key-size which consists in choosing codes with a large automorphism group, such as quasi-cyclic codes [64]. However, an structural algebraic attack [59] succeeds in breaking many of them. In 2012, Misoczki *et al.* in [105] proposed QC-MDPC with no algebraic structure which meet a very simple combinatorial property.

QC-MDPC Codes

A code C is called Quasi-Cyclic (QC) if there exists some integer n_0 such that every cyclic shift of a codeword $c \in C$ by n_0 positions give another codeword $c' \in C$. The main characteristics of QC-MDPC code are:

- For $n = n_0 p$ where p is an integer, both parity-check and generator matrix are composed by $n_0 p \times p$ circulant blocks.
- We just need to stock the first line of each circulant block to describe QC matrix.

Finally note that a (n, r, w) – MDPC code is an (n, k) binary linear code defined by a parity-check matrix with constant row weight w and a (n, r, w) – QC – MDPC code is a (n, r, w) – ModerateDensityParity – Check(MDPC) code that is quasi-cyclic with $n = n_0 r$.

Now, we describe the QC-MDPC version of McEliece cryptosystem presented in [105]. We present the key generation, the encryption and the decryption mechanisms.

Key generation. To generate a pair-key (t -error correcting (n, r, w) -QC-MDPC code where $n = n_0 r$) for the QC-MDPC version of McEliece cryptosystem we follow these steps:

1. Select the first rows $h_0, \dots, h_{n_0-1} \in \mathbb{F}_2^r$ of the n_0 parity-check matrix blocks $H_0, \dots, H_{n_0-1} \in \mathbb{F}_2^{r \times r}$ with weight $\sum_{i=0}^{n_0-1} wt(h_i) \leq w$ uniformly at random.

Chapter 2 Definitions and Background

2. Generate the parity-check matrix blocks $H_0, \dots, H_{n_0-1}$ by $r-1$ quasi-cyclic shifts of $h_0, \dots, h_{n_0-1}$.
3. The parity-check matrix H , which is the private key, is formed by concatenating $H_0, \dots, H_{n_0-1}$.
4. Generator matrix $G = [I_k | Q]$, which is the public key, is computed from H in row reduced echelon form by concatenating the identity matrix I_k with:

$$Q = \begin{pmatrix} (H_{n_0-1}^{-1} \cdot H_0)^T \\ (H_{n_0-1}^{-1} \cdot H_1)^T \\ \vdots \\ (H_{n_0-1}^{-1} \cdot H_{n_0-2})^T \end{pmatrix}$$

Encryption. To encrypt a message $m \in \mathbb{F}_2^k$, we follow just two steps:

1. Generate an error vector $e \in \mathbb{F}_2^n$ with at most t set bits uniformly at random.
2. Compute $x = m \cdot G + e$.

Decryption. To decrypt a ciphertext $x \in \mathbb{F}_2^n$, compute $mG \leftarrow R_H(x)$ using a t -error correcting QC-MDPC decoder R_H . Since G is of systematic form, extract m from the first k positions of $m \cdot G$.

We present the advantage of using QC-MDPC crypto-system as follow:

- Its security is based on the syndrome decoding problem which is resistant to quantum computer;
- It provides a compact public key size since we just need to stock the first line of our matrix generator;
- It is easy to implement and very fast since its implementation uses only bitwise and bit rotation operations.

To study the security of the QC-MDPC version of McEliece cryptosystem, we recall the following definition from [105].

Definition 7 (Codeword Finding Problem). *The $CwF(n, k, w)$ problem is as follows: given a matrix $H \in \mathbb{F}_2^{k \times n}$ and an integer w , the problem is to find a codeword of weight at most w in the code of parity-check matrix H .*

We also recall the following proposition from [105].

Proposition 3 ([105]). *Given the hardness of Codeword finding problem from MDPC and QC-MDPC codes:*

- *Breaking the MDPC variant of McEliece is not easier than solving the syndrome decoding problem for a random code.*
- *Breaking the QC-MDPC variant of McEliece is not easier than solving the syndrome decoding problem for a random quasicyclic linear code.*

Table 2.2: Some proposed QC-MDPC instances [105].

n	r	w	t	n_0	Key Size	Security
9602	4801	90	84	2	4801	80
19714	9857	142	143	2	9857	128
65542	32771	274	264	2	32771	256

In [105], Misoczki *et al.* proposed many parameters (see Tab 2.2) and recommended, for 80 bits of security, to use following parameters:

$$n_0 = 2, n = 9602, r = 4801, w = 90, t = 84. \quad (2.8)$$

In [77], Qian Guo *et al.* showed that it is possible to successfully recovers the proposed instance of QC-MDPC for 80 bit security secret key in a few minutes. So it is recommended to use the proposed instance of QC-MDPC for 128 bit security.

2.1.2 Rank metric

In coding theory, beside using the Hamming metric, we can also use the Rank metric. Let q be a power of a prime p and m an integer > 0 . We denote by $GL_m(\mathbb{F}_q)$ the group of invertible

$m \times m$ matrices over $\mathbb{F}_q$. Let $\beta = (\beta_1, \dots, \beta_m)$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Let $\mathcal{F}_i$ be the map from $\mathbb{F}_{q^m}$ to $\mathbb{F}_q$ where $\mathcal{F}_i(x)$ is the i^{th} coordinate of x in the basis β . For any $v = (v_1, \dots, v_n)$ in $\mathbb{F}_{q^m}^n$ we construct the matrix $\bar{v} \in \mathcal{M}_{m,n}(\mathbb{F}_q)$ where

$$\bar{v} = (\bar{v}_{i,j}) = \mathcal{F}_i(v_j) \quad (2.9)$$

The rank weight of a vector v denoted $\text{rank}(v)$ is defined as the rank of the associated matrix $\bar{v}$ defined by (2.9). For a given basis β , we denote by Φ_β the inverse computed in the basis β of the map

$$\mathbb{F}_{q^m}^n \rightarrow \mathcal{M}_{m,n}(\mathbb{F}_q) : v \mapsto \bar{v} \quad (2.10)$$

Like in Hamming metric, the minimum rank distance of a code C is the minimum rank weight of non null vectors of the code C . Loidreau showed in [99] that random rank-metric codes satisfy a Gilbert-Varshamov-like bound and that random codes can easily reach this bound.

Let $S(n, m, q, t)$ be the sphere of center 0 and radius t in $\mathbb{F}_{q^m}^n$ and $B(n, m, q, t)$ be the ball of center 0 and radius t in $\mathbb{F}_{q^m}^n$. The number of elements of $S(n, m, q, t)$ is given in [99] by:

$$\#S(n, m, q, t) = \prod_{j=0}^{t-1} \frac{(q^n - q^j)(q^m - q^j)}{q^t - q^j} \quad (2.11)$$

From (2.11), the volume of $B(n, m, q, t)$ is given by:

$$\#B(n, m, q, t) = \sum_{i=0}^t \#S(n, m, q, i) \quad (2.12)$$

Now we recall from [99] the bounds for rank metric codes.

Singleton Bound. The Singleton bound for a rank metric code C over $\mathbb{F}_{q^m}$ of length n , size M and minimal rank distance r is given by:

$$M \leq q^{\min(m(n-r+1), n(m-r+1))} \quad (2.13)$$

Rank Gilbert-Varshamov bound. The bound of Gilbert-Varshamov for rank metric codes is the conditions of existence of these ones. We give these conditions in Theorem 1.

Theorem 1 ([99]). *If $M \times \#B(n, m, q, r-1) < q^{nm}$ then there exists a code C of length n , size $M+1$ and minimal rank distance r .*

Properties of the Asterisk Product

In [25], T. Berger computed the isometry group for the rank distance using right multiplication by an invertible matrix over $\mathbb{F}_q$ and multiplication of columns by a non-zero element of $\mathbb{F}_{q^m}$. These families give the possibility to transform a word with fixed rank to any other word with the same rank. In Hamming metric, to transform a word with given weight to any other word with the same weight, we use a permutation. We now recall from [69] the definition of the asterisk product " $*$ ".

Definition 8 (Asterisk Product " $*$ "). *Let $Q \in \mathcal{M}_{m,n}(\mathbb{F}_q)$ and let $V(n)$ be an n dimensional vector space over the finite field $\mathbb{F}_{q^m}$, $v \in V(n)$ and β a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. We define the product $Q * v$ by*

$$Q * v = \Phi_\beta(Q \cdot \bar{v}),$$

where $\bar{v}$ is constructed from the basis β as in (2.9) and Φ_β as in (2.10).

Proposition 4 ([69]). *The asterisk product has the important following properties :*

1. *For all $x \in V(n)$ and $Q \in GL_m(\mathbb{F}_q)$, $\text{rank}(Q * x) = \text{rank}(x)$.*
2. *For all $x \in V(n)$, $P \in \mathcal{M}_{n,n}(\mathbb{F}_q)$ and $Q \in \mathcal{M}_{m,m}(\mathbb{F}_q)$, we have $(Q * x) \cdot P = Q * (x \cdot P)$.*
3. *For all $x, y \in V(n)$ and $\text{rank}(x) = \text{rank}(y)$, it is possible to find $P \in \mathcal{M}_{n,n}(\mathbb{F}_q)$ and $Q \in \mathcal{M}_{m,m}(\mathbb{F}_q)$ such that $x = Q * y \cdot P$.*

The General Decoding Problem in Rank Metric

The Syndrome Decoding Problem in Hamming metric (Problem 2) can be extended to the rank metric :

Problem 3 (Rank Syndrome Decoding (RSD) Problem [41]). *Let n, k and r be non negative integers, given a uniformly random matrix $H \in \mathcal{M}_{n-k \times n}(\mathbb{F}_{q^m})$ and a uniformly random syndrome $y \in \mathbb{F}_{q^m}^{(n-k)}$, find a vector $s \in \mathbb{F}_{q^m}^n$ such that $\text{rank}(s) = r$ and $H \cdot s^\top = y^\top$.*

Problem 4 (Rank General Decoding (RGD) Problem [41]). *Let n, k and r be non negative integers, given a uniformly random matrix $G \in \mathcal{M}_{k \times n}(\mathbb{F}_{q^m})$ and a uniformly random vector $x \in \mathbb{F}_{q^m}^n$, find vectors $e \in \mathbb{F}_{q^m}^n$ and $f \in \mathbb{F}_{q^m}^k$ such that $\text{rank}(e) = r$ and $f \cdot G \oplus e = x$.*

The Rank Syndrome Decoding Problem is proved to be difficult in [70] with a probabilistic reduction to the Hamming metric Syndrome Decoding Problem. In other words, if there were efficient probabilistic algorithms for solving the RSD problem, then there would exist efficient probabilistic algorithm to solve the Syndrome Decoding problem in Hamming metric. Furthermore, a recent combinatorial attack against RSD in [6] showed the average complexity is of

$$\mathcal{O}((n-k)^3 m^3 q^{r(k+1)m/n-m})$$

operations in $\mathbb{F}_q$, where parameters are as in Problem 3.

The McEliece Cryptosystem in Rank Metric

In [68], a McEliece-like cryptosystem using DC-LRPC codes is presented by Gaborit *et al.* A particularity of DC-LRPC codes is the fact that it is difficult to compute a parity check matrix (secret key) from the generator matrix (public key). We recall this cryptosystem as follow.

The keys generation step starts with a random choice of a DC-LRPC code C of length $n = 2k$, rank weight d and dimension k over $\mathbb{F}_{q^m}$, which can corrects errors of rank r . In a second time, we generate a parity check matrix $H = (A|B)$ which is a $k \times 2k$ double circulant matrix over $\mathbb{F}_{q^m}$ where A and B are $k \times k$ circulant matrices and A is invertible. Then, in a third time, we compute the code C generator matrix G as follow: $G = ((A^{-1}B)^\top | I_k)$ where I_k is the identity matrix of size k . Finally, the secret key is set to be H and the public key is set to be G . To encrypt a vector $x \in \mathbb{F}_{q^m}^k$, we choose a random vector $e \in \mathbb{F}_{q^m}^n$ of rank r and we compute the cipher text $c = x \cdot G + e$. To decrypt a cipher text $c \in \mathbb{F}_{q^m}^n$ we compute the syndrome $s = H \cdot c^T$ and recover the error vector e by decoding the DC-LRPC code and we compute $xG = c - e$ to recover then the vector x . Recently in [5], Aragon *et al.* proposed a new decoding algorithm which can correct $\frac{2}{3}(n-k)$ errors with a failure probability less than $q^{(1-r)(d-2)}$.

In Table 2.3, we give parameters of the McEliece-like cryptosystem using DC-LRPC codes.

Table 2.3: Some proposed DC-LRPC instances [68].

n	k	m	q	d	r	public key size(bits)	security (bits)
82	41	41	2	5	4	1681	80
106	53	53	2	6	5	2809	128
74	37	23	2^4	4	4	3404	110

2.2 Zero-Knowledge Identification Scheme

A Zero-knowledge identification scheme is an interactive protocol where one proves the knowledge of something, without revealing any information on this knowledge. This implies that the verifier cannot learn the secret, also he or she cannot replay the protocol and especially he or she cannot convince another person that the prover was aware of the secret. A zero-knowledge proof must satisfy three properties:

Completeness: if the statement is true, the honest verifier (that is, one following the protocol properly) will be convinced of this fact by a honest prover.

Soundness: if the statement is false, no cheating prover can convince the honest verifier that it is true, except with some small probability.

Zero-knowledge: if the statement is true, no verifier learns anything other than the fact that the statement is true. In other words, just knowing the statement (not the secret) is sufficient to imagine a scenario showing that the prover knows the secret. This is formalized by showing that every verifier has some simulator that, given only the statement to be proved (and no access to the prover), can produce a transcript that "looks like" an interaction between the honest prover and the verifier in question.

Zero-knowledge proofs was first introduced by Shafi Goldwasser, Silvio Micali, and Charles Rackoff in [74] where they introduced the hierarchy of interactive proof systems and the concept of knowledge complexity. They also gave the first zero-knowledge proof. Based on the instances of Syndrome Decoding problem, Stern presented in [123] a pioneering work that is the first code-based Zero-knowledge identification scheme whose cheating probability is $\frac{2}{3}$. In this scheme, the prover has to convince the verifier that he or she knows a word e of small weight such that $He^\top = y$ where the matrix $H \in \mathcal{M}_{(n-k) \times n}(\mathbb{F}_2)$ which is a parity check matrix of a linear binary code of length n and dimension k and the vector $y \in \mathbb{F}_2^{(n-k)}$ are public parameters. We recall the Stern identification scheme in Algorithm 2.1.

We also use AGS identification scheme [1] which is based on the Veron's Protocol [127]. Their scheme presented two improvements, the first one is about using double-circulant codes to increase the number of possible challenges, and the second one consists in a better use of commitments by compressing them. We recall this protocol in Algorithm 2.2. Let r be an

Algorithm 2.1 The Stern identification protocol[123]

1. The prover $\mathcal{P}$ randomly chooses a binary string u of length n and a permutation $\sigma \in \mathcal{S}_n$. He then sends c_1, c_2 and c_3 to $\mathcal{V}$ defined by:
 $c_1 = h(\sigma|Hu^\top); c_2 = h(\sigma(u)); c_3 = h(\sigma(u \oplus e))$ (where h is a hash function).
 2. $\mathcal{V}$ sends a random challenge $g \in \{0, 1, 2\}$ to $\mathcal{P}$.
 3. three possibilities occur:
 - a) if $g = 0$: $\mathcal{P}$ reveals u and σ .
 - b) if $g = 1$: $\mathcal{P}$ reveals $(u \oplus e)$ and σ .
 - c) if $g = 2$: $\mathcal{P}$ reveals $\sigma(u)$ and $\sigma(e)$.
 4. Verification step by the verifier:
 - a) if $g = 0$: $\mathcal{V}$ checks that c_1 and c_2 .
 - b) if $g = 1$: $\mathcal{V}$ checks that c_1 and c_3 received at Step 1 were correctly computed.
 - c) if $g = 2$: $\mathcal{V}$ checks that c_3 and c_2 received at Step 1 were correctly computed and that the weight of $\sigma(e)$ is exactly w .
-

integer such that $0 \leq r \leq k$, we denote by Rot_r the left shift rotation of r positions applied to vectors of $\mathbb{F}_2^k$, this map is applied by blocks of size k .

Private key: (e, f) with $e \in \mathbb{F}_2^n$ of Hamming weight w and f a random element of $\mathbb{F}_2^k$.

Public key: (G, x, w) with G a random matrix of size $k \times n$ and $x = e \oplus f \cdot G$.

Fiat-Shamir paradigm. It is possible to transform an identification protocol into a signature scheme through the Fiat-Shamir paradigm [62]. The general difference between authentication and signature is the number of characters involved. In fact, an authentication protocol consists of an interaction between a prover and a verifier while a signature needs only one signer. Fiat and Shamir proposed an idea consisting in the generation of the challenge with a random oracle. We explain in the following how it works: Firstly, a signer generates all the commitments at once. By applying an oracle to these elements, he obtains challenges that are used to compute the answers to send to the verifier. We use a hash function as a random oracle and the challenges are deduced from the hash value of the message and the commitments.

The Fiat-Shamir paradigm has been proved secure for the three-pass protocols in [111] and more recently for five-pass protocols in [4].

Algorithm 2.2 The AGS protocol [1]

1. A prover $\mathcal{P}$ randomly chooses $u \in \mathbb{F}_2^k$ and a permutation σ of $\mathcal{S}_n$. Then $\mathcal{P}$ sends to a verifier $\mathcal{V}$ the commitments c_1 and c_2 such that: $c_1 = h(\sigma)$ and $c_2 = h(\sigma(u \cdot G))$ where h is a hash function.
2. The verifier $\mathcal{V}$ sends $0 \leq r \leq k-1$ (the number of shifted positions) to $\mathcal{P}$.
3. The prover $\mathcal{P}$ builds $e_r = \text{Rot}_r(e)$, $f_r = \text{Rot}_r(f)$ and sends the last part of the commitments: $c_3 = h(\sigma(u \cdot G \oplus e_r))$.
4. The verifier $\mathcal{V}$ sends $g \in \{0, 1\}$ to $\mathcal{P}$:
5. Two possibilities:
 - if $g = 0$: $\mathcal{P}$ reveals $(u \oplus f_r)$ and σ .
 - if $g = 1$: $\mathcal{P}$ reveals $\sigma(u \cdot G)$ and $\sigma(e_r)$.
6. Verification Step, two possibilities:
 - if $g = 0$: $\mathcal{V}$ verifies that c_1, c_3 have been honestly computed;
 - if $g = 1$: $\mathcal{V}$ verifies that c_2, c_3 have been honestly computed and that $wt(\sigma(e_r)) = w$.

2.3 Group Signature

The fundamental cryptographic primitive which allows group users to anonymously sign documents on behalf of the whole group is called group signature, but in case of abuse, an administrator can revoke the anonymity of the signer. Group signature was introduced in [42]. Since then, many works have been proposed in this area [11, 30, 37], [75]. Group signature can be static, dynamic or fully dynamic.

2.3.1 Static Group Signature

Here, we are interested in the static group signature case. For that, we will follow the definitions presented in the BMW model [20].

Definition 9. A group signature scheme GS is a tuple of four polynomial time algorithms $(KeyGen, Sign, Verify, Open)$. The description of these algorithms is as follows:

1. *KeyGen* : this randomized algorithm takes as input $(1^\lambda; 1^N)$, where $N \in \mathbb{N}^*$ is the number of group users, and outputs $(g_{pk}, g_{msk}, g_{sk})$, where g_{pk} is the group public key, g_{msk} is the

group manager's secret key and $g_{sk} = \{g_{sk}[j]\}_{j \in [0, N-1]}$ with $g_{sk}[j]$ being the secret key for the group user of index j .

2. *Sign* : it is a randomized algorithm, it takes as input a secret signing key $g_{sk}[j]$ for some $j \in [0, N-1]$ and a message M and returns a group signature Σ on M .
3. *Verify* : it is a deterministic algorithm that takes as input the group public key g_{pk} , a message M , a signature Σ on M , and returns either 1 (Accept) or 0 (Reject).
4. *Open* : this deterministic algorithm takes as input the group manager's secret key g_{msk} , a message M , a signature Σ on M and returns an index $j \in [0, N-1]$ associated with a particular user, or False indicating failure.

A group signature scheme as described in BMW model [20] must verify three security requirements:

- *Correctness*: for all integers λ and N , all $(g_{pk}, g_{msk}, g_{sk})$ obtained from *KeyGen* algorithm with $(1^\lambda; 1^N)$ as input, all $j \in \{0, \dots, N-1\}$ and $M \in \{0, 1\}^*$

$$\text{Verify}(g_{pk}, M, \text{Sign}(g_{pk}, g_{sk}[j], M)) = 1$$

and

$$\text{Open}(g_{pk}, g_{msk}, M, \text{Sign}(g_{pk}, g_{sk}[j], M)) = j$$

- *Traceability*: requires that all signatures, even those produced by a coalition of group users and the group manager, can be traced back to a member of the coalition.
- *Anonymity*: requires that, signatures generated by two distinct group users are computationally indistinguishable to an adversary knowing all the user secret keys.

Formal definitions of these properties are described in the following definitions:

Definition 10 (CPA anonymity). *We say that a group signature $GS = (\text{KeyGen}, \text{Sign}, \text{Verify}, \text{Open})$ is CPA-anonymous if for all polynomial $N(\cdot)$ and any Probabilistic Polynomial Time (PPT) adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following experiment is negligible in λ :*

1. Run $(g_{pk}, g_{msk}, g_{sk}) \leftarrow \text{KeyGen}(1^\lambda, 1^N)$ and send (g_{pk}, g_{sk}) to $\mathcal{A}$.

2. $\mathcal{A}$ outputs two identities $j_0, j_1 \in [0, N-1]$ with a message M . Choose a random bit b and give $\text{Sign}(g_{sk}[j_b], M)$ to $\mathcal{A}$. Then, $\mathcal{A}$ outputs a bit b_0 . $\mathcal{A}$ succeeds if $b_0 = b$, and the advantage of $\mathcal{A}$ is defined to the modulus of $\Pr[\mathcal{A} \text{ succeeds}] - \frac{1}{2}$.

Definition 11 (Traceability). We say that a group signature $GS = (\text{KeyGen}, \text{Sign}, \text{Verify}, \text{Open})$ is traceable if for all polynomial $N(\cdot)$ and any PPT adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following experiment is negligible in λ :

1. Run $(g_{pk}, g_{msk}, g_{sk}) \leftarrow \text{KeyGen}(1^\lambda, 1^N)$ and send (g_{pk}, g_{msk}) to $\mathcal{A}$.
2. $\mathcal{A}$ may query the following oracles adaptively and in any order:
 - A O^{Corrupt} oracle that on input $j \in [0, N-1]$, outputs $g_{sk}[j]$.
 - A O^{Sign} oracle that on input j , a message M , returns $\text{Sign}(g_{sk}[j], M)$.

Let CU be the set of identities queried to O^{Corrupt} .

3. Finally, $\mathcal{A}$ outputs a message M^* and a signature Σ^* .

$\mathcal{A}$ succeeds if (1) $\text{Verify}(g_{pk}, M^*, \Sigma^*) = 1$ and (2) $\text{Sign}(g_{sk}[j], M^*)$ was never queried for $j \notin CU$, yet (3) $\text{Open}(g_{msk}, M^*, \Sigma^*) \notin CU$.

2.3.2 Dynamic Group Signature

In this section, we are interested in the dynamic group signature case which is very useful in many applications such as sybil attack detection in a vehicular ad hoc network and privacy preserving data mining. We follow the definitions presented in [29].

Definition 12. A group signature scheme $GS = (\text{Setup}, \text{Join}, \text{Sign}, \text{Verif}, \text{Open})$ is a sequence of protocols such as:

- $\text{Setup}(1^\lambda)$: this algorithm generates global public parameters of the system params, the group public key g_{pk} and the group manager secret key g_{msk} encompassing the opening key, s_{kO} ;

- $Join(\mathcal{U}_i)$: this is an interactive protocol between a user $\mathcal{U}_i$ and the group manager. At the end of the protocol, the user obtains a secret signing key $s_k[i]$. The group manager adds the new user $\mathcal{U}_i$ and updates s_{kO} ;
- $Sign(g_{pk}, s_k[i], m; \mu)$: to sign a message m , the user uses his secret key $s_k[i]$ and some randomness μ to output a signature σ valid under the group public key g_{pk} ;
- $Verif(g_{pk}, m, \sigma)$: anybody should be able to verify the validity of the signature σ on the message m with respect to g_{pk} . It thus outputs 1 if the signature is valid, and 0 otherwise;
- $Open(s_{kO}, g_{pk}, m, \sigma)$: for a valid signature σ with respect to g_{pk} , the group manager can provide the signer identity. It thus outputs the user $\mathcal{U}_i$.

Following [29], to be claimed secure, a group signature scheme has to prove its correctness and fulfill three properties: anonymity(*anon*), traceability(*tr*) and non frameability(*nf*). For more details on security requirement of group signature, see [29].

Correctness: this notion guarantees that honest users should be able to generate valid signatures and the opener should then be able to revoke anonymity of the signers.

Unforgeability: informally, the unforgeability notion guarantees that no one can produce a valid signature that cannot be opened in convincing way (traceability) and that no one can produce a signature on behalf of some group member (non-frameability).

In the following experiments, to join the group, an adversary($\mathcal{A}$) runs the joinP-oracle (passive join). $\mathcal{A}$ creates a honest user for whom he does not know the secret key: the index i is added to the *HU* (Honest Users) list.

For users whose secret keys are known to the adversary, we let the adversary play on their behalf. For honest users, the adversary can interact with them granted some oracles:

- $corrupt(i)$, if $i \in HU$, provides the secret key $s_k[i]$ of this user. The adversary can now control it. The index i is then moved from *HU* to the list of corrupted users *CU*;
- $sign(i, m)$, if $i \in HU$, plays as the honest user $\mathcal{U}_i$ would do in the signature process. Then i is appended to the list $S[m]$.

We also define the open-oracle which on input (m, σ) returns $Open(s_{kO}, g_{pk}, m, \sigma)$.

Algorithm 2.3 Unforgeability Notions [29](a) Experiment $Exp_{GS,\mathcal{A}}^{tr}(\lambda)$

1. $(g_{pk}, g_{msk}, s_{kO}) \leftarrow Setup(1^\lambda)$
2. $(m, \sigma) \leftarrow \mathcal{A}(g_{pk} : joinP, corrupt, sign, open)$
3. if $Verif(g_{pk}, m, \sigma) = 0$, return 0
4. if $\exists j \notin CU \cup S[m]$,
 $Open(g_{msk}, g_{pk}, m, \sigma) = j$
return 1
else return 0
 $Adv_{GS,\mathcal{A}}^{tr}(\lambda) = Pr[Exp_{GS,\mathcal{A}}^{tr}(\lambda) = 1]$

(b) Experiment $Exp_{GS,\mathcal{A}}^{nf}(\lambda)$

1. $(g_{pk}, g_{msk}, s_{kO}) \leftarrow Setup(1^\lambda)$
2. $(m, \sigma) \leftarrow \mathcal{A}(g_{pk}, g_{msk} : joinP, corrupt, sign, open)$
3. if $Verif(g_{pk}, m, \sigma) = 0$, return 0
4. if $\exists i \in HU \setminus S[m]$, $Open(g_{msk}, g_{pk}, m, \sigma) = i$
return 1
else return 0
 $Adv_{GS,\mathcal{A}}^{nf}(\lambda) = Pr[Exp_{GS,\mathcal{A}}^{nf}(\lambda) = 1]$

Traceability and Non-Frameability

Traceability (see Algorithm 2.3(a)) says that nobody should be able to produce a valid signature that cannot be opened in a convincing way. Furthermore, non-frameability (see Algorithm 2.3(b)) guarantees that no dishonest player (even the authorities, i.e. the Group Manager $\mathcal{GM}$, hence the keys when g_{msk} is provided to the adversary) will be able to frame a honest user: a honest user that does not sign a message m should not be convincingly declared as a possible signer. Non-frameability also shows that the group manager cannot cheat. Thus, we say that:

- GS is traceable if, for any polynomial adversary $\mathcal{A}$, the advantage $Adv_{GS,\mathcal{A}}^{tr}(\lambda)$ is negligible.
- GS is non-frameable if, for any polynomial adversary $\mathcal{A}$, the advantage $Adv_{GS,\mathcal{A}}^{nf}(\lambda)$ is negligible.

In both games, the adversary generates a signature σ on a message m of its choice. In the latter game, the adversary itself can play the role of the opener trying to spoof a honest user i .

Anonymity

Given two of honest users i_0 and i_1 , the adversary should not have any significant advantage in guessing which one of them have issued a valid signature.

Algorithm 2.4 Anonymity notion

Experiment $Exp_{GS, \mathcal{A}}^{anon-b}(\lambda)$

1. $(g_{pk}, g_{msk}, s_{kO}) \leftarrow (1^\lambda)$
2. $(m, i_0, i_1) \leftarrow \mathcal{A}(FIND, g_{pk} : joinP, corrupt, sign)$
3. $\sigma \leftarrow Sign(g_{pk}, i_b, m, s_k[i])$
4. $b' \leftarrow \mathcal{A}(GUESS, \sigma : joinP, corrupt, sign)$
5. if $i_0 \notin HU$ or $i_1 \notin HU$ return 0
6. return b'

$$Adv_{GS, \mathcal{A}}^{anon}(\lambda) = Pr[Exp_{GS, \mathcal{A}}^{anon-1}(\lambda) = 1] - Pr[Exp_{GS, \mathcal{A}}^{anon-0}(\lambda) = 1]$$

The adversary can interact with honest users as before (with sign and corrupt) but the challenge signature is generated using the interactive signature protocol $Sign$. The adversary plays the role of corrupted users whereas honest users are activated to play their roles.

GS is anonymous if, for any polynomial adversary $\mathcal{A}$, the advantage $Adv_{GS, \mathcal{A}}^{anon}(\lambda)$ is negligible.

2.3.3 Fully Dynamic Group Signature

In this section, we are interested in the dynamic group signature case. For this reason, we follow the definitions presented in [31].

Definition 13. A *Fully Dynamic Group Signature (FDGS)* is a tuple of polynomial-time algorithm described as follow:

GSetup (1^λ) This algorithm generates global public parameters of the system p_p .

$\langle \mathbf{GKgen}_{GM}(p_p), \mathbf{GKgen}_{TM}(p_p) \rangle$ In this interactive protocol ran the group manager GM and the tracing manager, algorithm $\mathbf{GKgen}_{GM}$ outputs a manager public and private keys

(m_{pk}, m_{sk}) . At the same time, GM initializes the group information $\mathcal{J}$ and the registration table $\mathbf{r}$. The algorithm GKgen_{TM} outputs the tracing public and private keys (t_{pk}, t_{sk}) . At the end the group public key is $g_{pk} = (p_p, m_{pk}, t_{pk})$.

UKgen(p_p) This algorithm generate a user public and private keys (u_{pk}, u_{sk}) .

$\langle \text{Join}(\mathcal{J}_{\tau_{\text{current}}}, g_{pk}, u_{pk}, u_{sk}); \text{Issue}(\mathcal{J}_{\tau_{\text{current}}}, m_{sk}, u_{id}) \rangle$ In this interactive protocol run by the user and GM, the algorithm Join is used to add a user as group member and to store private group signing key $g_{sk}[u_{id}]$. While the algorithm Issue is used to store registration information in the table $\mathbf{r}$ with index u_{id} .

GUpdate($g_{pk}, m_{sk}, \mathcal{J}_{\tau_{\text{current}}}, \mathbb{S}, \mathbf{r}$) In this algorithm ran by the GM, the group information is updated while advancing the epoch. Given $g_{pk}, m_{sk}, \mathcal{J}_{\tau_{\text{current}}}$, registration table $\mathbf{r}$, a set $\mathbb{S}$ of active users to be removed from the group, GM computes new group information $\mathcal{J}_{\tau_{\text{new}}}$ and may update the table $\mathbf{r}$.

IsActive($\mathcal{J}_{\tau}, \mathbf{r}, u_{id}$) This algorithm outputs 1 if this user is active at epoch τ and 0 otherwise.

Sign($g_{pk}, g_{sk}(u_{id}), \mathcal{J}_{\tau}, M$) This algorithm outputs a group signature Σ on message M by user u_{id} .

Verify($g_{pk}, \mathcal{J}_{\tau}, M, \Sigma$) This algorithm checks the validity of the signature Σ on message M at epoch τ .

Trace($g_{pk}, t_{sk}, \mathcal{J}_{\tau}, \mathbf{r}, M, \Sigma$) This is an algorithm run by TM. Given the inputs, TM returns an identity u_{id} of a group member who signed the message and a proof indicating this tracing result Π_{trace} .

Judge($g_{pk}, u_{id}, \mathcal{J}_{\tau}, \Pi_{\text{trace}}, M, \Sigma$) This algorithm checks the validity of Π_{trace} outputted by the Trace algorithm.

Following [31], An FDGS is secure if it is correct, anonymous, traceable and non-frameable as defined below.

Before starting to define these property we need to introduce some oracles. In these oracles the following global variables: $\mathbb{H}$ is a set of users whose keys are generated honestly. $\mathbb{B}$ is a list of users whose secret signing keys are revealed to the adversary; $\mathbb{D}$ is a list of users whose public

Chapter 2 Definitions and Background

keys are chosen by the adversary; $\mathbb{S}_g$ is a set of signatures that are generated by the oracle GenSign ; $\mathbb{S}_c$ is a set of signatures that are generated by the oracle Chal_b .

AddHU(u_{id}) This oracle adds a single honest user u_{id} to the group. It output the user public key $u_{pk}[u_{id}]$ and adds u_{id} to the list $\mathbb{H}$.

CrptU(u_{id}, p_k) In this oracle the adversary $\mathcal{A}$ can create a new user by choosing the key $u_{pk}[u_{id}]$ as p_k . It also adds u_{id} to the list $\mathbb{D}$. It returns null if $u_{id} \in \mathbb{H} \cup \mathbb{D}$.

SndToM(u_{id}, M) In this oracle the $\langle \text{Join}, \text{Issue} \rangle$ protocol is engaged between the adversary $\mathcal{A}$, who corrupts the group manager u_{id} in the list $\mathbb{D}$, and the honest Issue-executing group manager.

SndToU(u_{id}, M) In this oracle the $\langle \text{Join}, \text{Issue} \rangle$ protocol is engaged between the adversary $\mathcal{A}$, who corrupts the group manager, and the honest Join-executing user u_{id} in the set $\mathbb{H}$.

ReadReg(u_{id}) given a session identifier u_{id} , this oracle returns the corresponding entry in the registry $r[u_{id}]$.

ModReg(u_{id}, ρ) This oracle modifies $r[u_{id}]$ to any ρ chosen by the adversary $\mathcal{A}$.

RevealU(u_{id}) This oracle returns the user u_{id} secret key $g_{sk}[u_{id}]$ to the adversary $\mathcal{A}$, and adds this user to the list $\mathbb{H}$.

GenSign(u_{id}, M, τ) This oracle returns a signature on the message M by the user u_{id} at epoch τ , and then add the signature to the list $\mathbb{H}$.

Chal_b($\mathcal{I}_\tau, u_{id0}, u_{id1}, M$) This oracle takes as input some group information $\mathcal{I}$, a message m , and two honest users u_{id0}, u_{id1} sent by the adversary $\mathcal{A}$. It returns a group signature on the message using key g_{sk_b} for $b \in \{0, 1\}$ and adds the signature to the set $\mathbb{S}_c$.

GenTrace($M, \Sigma, \mathcal{I}_\tau$) This oracle returns the signer of the signature together with a proof, with respect to the epoch τ . We require that the signature is not in the set $\mathbb{S}_c$.

GenUpdate($\mathbb{S}$) In this oracle, the Adversary $\mathcal{A}$ can update the group at current epoch $\tau_{current}$. Here $\mathbb{S}$ is a set of active users at current epoch.

Correctness Correctness guarantees that if a honest user joins the group and produce a signature, this signature will be accepted by the Verify algorithm. In the $Exp_{FDGS,\mathcal{A}}^{corr}(\lambda)$ experiment (Algorithm 2.5), the adversary $\mathcal{A}$ can access to the AddHU, ReadReg and GenUpdate oracles. An FDGS is correct if for any PPT adversary $\mathcal{A}$, $Pr[Exp_{FDGS,\mathcal{A}}^{corr}(\lambda) = 1]$ is close to 1.

Algorithm 2.5 Experiment $Exp_{FDGS,\mathcal{A}}^{corr}(\lambda)$

Require: A security parameter λ .

Ensure: A value 0 or 1.

$p_p \leftarrow GSetup(\lambda)$, $\mathbb{H} = \emptyset$.

$\langle (\mathcal{J}, m_{pk}, m_{sk}); (t_{pk}, t_{sk}) \rangle \leftarrow \langle GKgen_{GM}(p_p), GKgen_{TM}(p_p) \rangle$.

Set $g_{pk} = (p_p, m_{pk}, t_{pk})$.

The adversary $\mathcal{A}$ using g_{pk} and $\mathcal{J}$, with access to AddHU, ReadReg and GenUpdate oracles, generate the tuple (u_{id}, M, τ)

if $u_{id} \notin \mathbb{H}$ or $g_{sk}[u_{id}] = null$ or $\mathcal{J}_\tau = null$ or $IsActive(\mathcal{J}_\tau, r, u_{id}) = 0$ **then**
 return 0.

end if

$\Sigma \leftarrow Sign(g_{pk}, g_{sk}(u_{id}), \mathcal{J}_\tau, M)$, $(u'_{id}, \Pi_{trace}) \leftarrow Trace(g_{pk}, t_{sk}, \mathcal{J}_\tau, r, M, \Sigma)$

if $Verify(g_{pk}, \mathcal{J}_\tau, M, \Sigma) = 0$ or $Judge(g_{pk}, u_{id}, \mathcal{J}_\tau, \Pi_{trace}, M, \Sigma) = 0$ or $u_{id} \neq u'_{id}$ **then**
 return 1.

end if

Anonymity Anonymity guarantees that it is infeasible for any PPT adversary to distinguish signatures generated by two active users of its choice at the chosen epoch, even if it can corrupt any user, can choose the key of GM, and is given access to the GenTrace oracle. The anonymity experiment $Exp_{FDGS,\mathcal{A}}^{anon-b}(\lambda)$ for $b \in \{0, 1\}$ is described in Algorithm 2.6.

Algorithm 2.6 Experiment $Exp_{FDGS,\mathcal{A}}^{anon-b}(\lambda)$

Require: A security parameter λ .

Ensure: A value 0 or 1.

$p_p \leftarrow GSetup(\lambda)$, set $\mathbb{H}, \mathbb{D}, \mathbb{B}, \mathbb{S}_g$ and $\mathbb{S}_c$ to $\emptyset$.

The adversary $\mathcal{A}$ uses $GKgen_{GM}$ and p_p to generate the tuple $(\mathcal{J}, m_{pk}, m_{sk})$.

if $GKgen_{GM}$ did not accept or $\mathcal{A}'$ output is not well-formed **then**
 return 0.

end if

set g_{pk} to (p_p, m_{pk}, t_{pk})

The adversary $\mathcal{A}$ using g_{pk} , with access to AddHU, CrptU, RevealU, SndToU, GenTrace, ModReg and $Chal_b$ oracles, outputs a bit b' .

return b' .

Non-Frameability guarantees that the adversary cannot generate a valid signature that is associate to a honest user even if it can corrupt all other users, and can choose keys of group

and tracing managers. The non-frameability experiment $Exp_{FDGS, \mathcal{A}}^{non-frame}(\lambda)$ is described in Algorithm 2.7.

Algorithm 2.7 Experiment $Exp_{FDGS, \mathcal{A}}^{non-frame}(\lambda)$

Require: A security parameter λ .

Ensure: A value 0 or 1.

$p_p \leftarrow GSetup(\lambda)$, set $\mathbb{H}$, $\mathbb{D}$, $\mathbb{B}$ and $\mathbb{S}_g$ to $\emptyset$.

The adversary $\mathcal{A}$ uses p_p to generate the tuple $(\mathcal{I}, m_{pk}, m_{sk}, t_{pk}, t_{sk})$.

if $\mathcal{A}'$ output is not well-formed **then**

return 0.

else

 Set g_{pk} to (p_p, m_{pk}, t_{pk}) .

end if

The adversary $\mathcal{A}$ using g_{pk} , with access to CrptU, RevealU, SndToU, ModReg and Sign oracles, outputs a tuple $(M, \Sigma, u_{id}, \Pi_{trace}, \mathcal{I}_\tau)$.

if $Verify(g_{pk}, \mathcal{I}_\tau, M, \Sigma) = 1$ and $Judge(g_{pk}, u_{id}, \mathcal{I}_\tau, \Pi_{trace}, M, \Sigma) = 1$ and $u_{id} \in \mathbb{H} \setminus \mathbb{B}$ and $(M, \Sigma, \tau) \notin \mathbb{S}_g$ **then**

return 1.

end if

Traceability ensures that the adversary cannot produce a valid signature that cannot be traced to an active user at the chosen epoch, even if it can corrupt any user and can choose the key of TM. The traceability experiment $Exp_{FDGS, \mathcal{A}}^{trace}(\lambda)$ is described in Algorithm 2.8.

Tracing Soundness requires that it is infeasible to produce a valid signature that traces to two different users, even if all groups user and both managers are fully controlled by the adversary. The tracing soundness $Exp_{FDGS, \mathcal{A}}^{trace-sound}(\lambda)$ is described in Algorithm 2.9.

2.4 Cryptographic Accumulator

The cryptographic accumulator notion was first introduced by Benaloh and De Mare [24] in 1993. The cryptographic accumulator is a cryptographic primitive which allows accumulating a finite set of elements into a single value that is called *Accumulated Value*. Therefore, for each element, it is possible to generate a short membership proof called Witness which attests the belonging of the element to the set. The accumulated value is public. For a cryptographic accumulator to be secure, it should not be possible to find a valid witness for an element not belonging to the set. In the classical example, to accumulate a finite set $X = \{x_i\}$ of big prime

Algorithm 2.8 Experiment $Exp_{FDGS, \mathcal{A}}^{trace}(\lambda)$ **Require:** A security parameter λ .**Ensure:** A value 0 or 1. $p_p \leftarrow GSetup(\lambda)$, set $\mathbb{H}, \mathbb{D}, \mathbb{B}$ and $\mathbb{S}_g$ to $\emptyset$.The adversary $\mathcal{A}$ uses $GKgen_{GM}$ and p_p to generate the tuple (t_{pk}, t_{sk}) .**if** $GKgen_{GM}$ did not accept or $\mathcal{A}'$ output is not well-formed **then****return** 0.**end if**set g_{pk} to (p_p, m_{pk}, t_{pk}) .The adversary $\mathcal{A}$ using g_{pk} , with access to AddHU, CrptU, RevealU, SndToM, ModReg, GenUpdate and Sign oracles, outputs a tuple (M, Σ, τ) .**if** $Verify(g_{pk}, \mathcal{I}_\tau, M, \Sigma) = 0$ **then****return** 0.**end if** $(u_{id}, \Pi_{trace}) \leftarrow Trace(g_{pk}, t_{sk}, \mathcal{I}_\tau, r, M, \Sigma)$.**if** $IsActive(\mathcal{I}_\tau, r, u_{id}) = null$ or $Judge(g_{pk}, u_{id}, \mathcal{I}_\tau, \Pi_{trace}, M, \Sigma) = 0$ or $u_{id} = 0$ **then****return** 1.**end if****Algorithm 2.9** Experiment $Exp_{FDGS, \mathcal{A}}^{trace-sound}(\lambda)$ **Require:** A security parameter λ .**Ensure:** A value 0 or 1. $p_p \leftarrow GSetup(\lambda)$, set $\mathbb{D}$ to $\emptyset$.The adversary $\mathcal{A}$ uses $GKgen_{GM}$ and p_p to generate the tuple $(\mathcal{I}, m_{pk}, m_{sk}, t_{pk}, t_{sk})$.**if** $\mathcal{A}'$'s output is not well-formed **then****return** 0.**else**Set g_{pk} to (p_p, m_{pk}, t_{pk}) .**end if**The adversary $\mathcal{A}$ using g_{pk} , with access to CrptU and ModReg oracles, outputs a tuple (M, Σ, τ) .**if** $Verify(g_{pk}, \mathcal{I}_\tau, M, \Sigma) = 1$ and $u_{id0} \neq u_{id1} \neq null$ and $Judge(g_{pk}, u_{idb}, \mathcal{I}_\tau, \Pi_{trace,b}, M, \Sigma) = 0$ or $u_{id} = 0$ for $b \in \{0, 1\}$ **then****return** 1.**end if**

integers, we compute the product x of all elements of X . To prove the membership of an element x_i of X , we just compute $q_i = \frac{x}{x_i}$ where q_i is called the witness of x_i . Then everyone can check the membership of x_i by checking if $x_i q_i = x$ which is an easy computation and q_i is an integer. Cryptographic accumulators are very useful in many privacy-preserving technologies like anonymous credential systems, group signatures, efficient time-stamping, sanitizable signatures, ring signatures, redactable signatures, identity-based encryption, Zerocoin, *etc.*...

Cryptographic accumulator can be static or dynamic, in this thesis we focus on the static case. We recall from [53] the definition and security requirements for a static cryptographic accumulator.

Definition 14 (Static Accumulator). *A static accumulator is a quadruple of efficient algorithms $(\text{Gen}, \text{Eval}, \text{WitCreate}, \text{TVerify})$ defined as follows:*

- $\text{Gen}(1^\lambda)$. *This algorithm takes a security parameter λ and returns a public key p_k .*
- $\text{Eval}(p_k, \mathcal{X})$. *This algorithm takes public key p_k and a set $\mathcal{X}$ to be accumulated in $A_{\mathcal{X}}$ and returns an accumulator $A_{\mathcal{X}}$ together with some auxiliary information ξ .*
- $\text{WitCreate}(p_k, \mathcal{X}, A_{\mathcal{X}}, \xi, x_i)$. *This algorithm takes public key p_k , the accumulated set $\mathcal{X}$, an accumulator $A_{\mathcal{X}}$, auxiliary information ξ and a value x_i . It returns error, if $x_i \notin \mathcal{X}$, and a witness W_{x_i} for x_i , otherwise.*
- $\text{TVerify}(p_k, A_{\mathcal{X}}, W_{x_i}, x_i)$. *This algorithm takes a public key p_k , an accumulator $A_{\mathcal{X}}$, a witness W_{x_i} and a value x_i . It returns true if W_{x_i} is a witness for $x_i \in \mathcal{X}$ and false otherwise.*

Remark 2. *We recall that a universal cryptographic accumulator allows proving both membership and non-membership. While in a non-universal cryptographic accumulator we can just prove either membership or non-membership.*

According to D. Derler *et al.* [53] to be clammed secure, a non-universal cryptographic accumulator must first be correct and prove its collision freeness and its indistinguishability which we define hereafter.

Correctness: This notion guarantees that for all honestly generated keys, all honestly computed accumulators and witnesses, the $\text{TVerify}()$ algorithm will always return true.

Collision Freeness: Informally, this notion states that it is neither feasible to find a witness for a non-accumulated value. In Definition 15, we will give a formal definition of Collision Freeness for static and non-universal cryptographic accumulator. A cryptographic accumulator is secure if it is correct and collision-free.

Definition 15 (Collision Freeness). *Let α be the output of Algorithm 2.10. A static and non-universal cryptographic accumulator is a collision-free, if for all probabilistic polynomial-*

time (PPT) algorithm of an adversary $\mathcal{A}$, the probability α of Collision Experiment (Algorithm 2.10) is negligible.

Algorithm 2.10 Collision Experiment

Require: The security parameter λ .

Ensure: The probability α of collision.

1. $p_k \leftarrow \text{Gen}(1^\lambda)$, (where $\leftarrow$ is the affectation symbol.)
2. $\mathcal{O} \leftarrow \{\mathcal{O}^E, \mathcal{O}^W\}$
3. The adversary $\mathcal{A}^{\mathcal{O}}$ with the access to p_k generates the tuples $(W_{x_i}, x_i, \mathcal{X})$

$$\alpha = \Pr[\text{TVerify}(p_k, A_{\mathcal{X}}, W_{x_i}, x_i) = \text{true and } x_i \notin \mathcal{X}]$$

Notation: In the following, $\mathcal{O}^E$ represents an oracle for the algorithm Eval() and an adversary $\mathcal{A}$ that have access to an oracle $\mathcal{O}$ is noted $\mathcal{A}^{\mathcal{O}}$. An adversary is allowed to query them an arbitrary number of times. The oracle $\mathcal{O}^W$ allows the adversary to obtain membership witnesses for members. In Collision Freeness Experiment (Algorithm 2.10), $A_{\mathcal{X}}$ is the execution result of Eval($p_k, \mathcal{X}$) and the adversary $\mathcal{A}$ has oracle access to $\mathcal{O}^E$ and $\mathcal{O}^W$. In Indistinguishability Experiment (Algorithm 2.11), $\mathcal{X}_0$ and $\mathcal{X}_1$ are two distinct subsets of the accumulation domain and *state* is additional informations on $\mathcal{X}_0$ and $\mathcal{X}_1$. We define $1^\lambda = \overbrace{11 \dots 1}^{\lambda \text{ times}}$ as a string of ones λ times where λ is a security parameter and $\Pr[A]$ is the probability to have the event A realized.

The notion of indistinguishability is that, given two different sets of values and an accumulated value, an adversary cannot decide from which accumulator comes the accumulated value. In Definition 16, we give a formal definition of indistinguishability for static and non-universal cryptographic accumulator.

Definition 16 (Indistinguishability). A static and non-universal cryptographic accumulator is indistinguishable, if for all probabilistic polynomial-time (PPT) algorithm of an adversary $\mathcal{A}$, the probability β of Indistinguishability Experiment given by Algorithm 2.11 is close to $\frac{1}{2}$.

If the probability β is exactly $\frac{1}{2}$, we have unconditional indistinguishability, whereas we have computational indistinguishability if the probability is negligibly different from $\frac{1}{2}$.

Algorithm 2.11 Indistinguishability Experiment

Require: The security parameter λ .

Ensure: The probability β of Indistinguishability Experiment.

1. $p_k \leftarrow \text{Gen}(1^\lambda)$
2. pick a random $b \in \{0, 1\}$
3. $(\mathcal{X}_0, \mathcal{X}_1, \text{state}) \leftarrow \mathcal{A}(p_k)$, the adversary $\mathcal{A}$ with the access to p_k , generates a tuple $(\mathcal{X}_0, \mathcal{X}_1, \text{state})$
4. $(A_{\mathcal{X}_b}, \xi) \leftarrow \text{Eval}(p_k, \mathcal{X}_b)$
5. $\mathcal{O} \leftarrow \{\mathcal{O}^E, \mathcal{O}^W\}$
6. The adversary $\mathcal{A}^\mathcal{O}$ with the access to $(p_k, A_{\mathcal{X}_b}, \text{state})$, generates a bit $g \in \{0, 1\}$

$$\beta = \Pr[b = g]$$

2.5 Conclusion

In this chapter we have presented some definitions and background in code-based cryptography, zero-knowledge identification scheme, group signature scheme and cryptographic accumulators. These notions presented above are needed for the well understanding of the remaining chapters.

A Code-based Group Signature Scheme with Shorter Public Key Length

Contents

3.1	Introduction	43
3.2	Preliminaries	45
3.2.1	Notations	45
3.2.2	Code-based Cryptography Background	45
3.3	Our Proposed Group Signature Scheme	47
3.4	Security Analysis	50
3.5	Results	52
3.6	Conclusion	52

In this chapter based essentially on [8], we show that it is possible to reduce the public key length of the first provably secure group signature scheme from code-based assumptions without losing the security properties. More precisely, the public key can be 466 times shorter than the original scheme, typically for a group of 16 users when the public key length is 1.34 kilo-bytes, while the size is 625 kilo-bytes in the original scheme [58]. Our technic consist in using a Quasi-cyclic Moderate Density Parity-Check McEliece variant for encrypting user identity and a random double circulant matrix for the Underlying Zero Knowledge Argument System.

3.1 Introduction

A fundamental cryptographic primitive which allows group users to anonymously sign documents on behalf of the whole group is called group signature, but in case of abuse, an administrator can revoke the anonymity of the signer. Group signature was introduced in [42]. Since then, many works have been proposed in this area [11, 30, 37, 75]. For anonymity and traceability properties of these schemes, group signatures are highly useful in various real-life scenarios such as controlled anonymous printing services, digital right management systems, e-bidding

and e-voting schemes. The majority of group signature schemes are based on number theory assumptions [30, 32, 38, 94], but number-theoretic based cryptography will not resist to the quantum computing. Recently, the research for post-quantum group signatures is quite active as shows these publications [3, 75, 89, 90, 95]. The majority of these works are based on lattice assumptions, while in code-based cryptography we denote two group signature schemes: the first one is based on BSZ model [22] and it is presented in [3]. The second one is based on BMW model [20] and is presented in [58], this scheme satisfies the CPA-anonymity and traceability requirements in the random oracle model. But the size of the signature and the public key makes this scheme impractical.

In this chapter, we propose a new provably secure group signature scheme based on code assumptions presented in [58]. In this construction we replace the original McEliece cryptosystem used to encrypt the signer identity by the Quasi-cyclic Moderate Density Parity-Check (QC-MDPC) version of McEliece. The advantage of QC-MDPC codes is that for the same security level, the QC-MDPC version of McEliece has very short public key size than the original version. For example, for 80 bits security level, the QC-MDPC public key is 4801 bits unlike the original McEliece public key which is around 500000 bits. The second improvement consist in using a random double circulant matrix for the Underlying Zero Knowledge Argument System as proposed in [65], we suggest to replace a random matrix used by Stern [124] in the Zero-Knowledge identification scheme by a random double circulant matrix. This construction allows to have a very short public key, only 349 bits to obtain 2^{83} of security level in the Zero-Knowledge identification scheme. These two improvements reduce dramatically the public key size of the group signature scheme without impacting the security level. For instance, in the case of 16 users we achieve a public key size of 1.34 KB, while the size is around 625 KB in the original scheme [58]. Moreover, the structure of these matrices makes the implementation easier and efficient, since the product of a circulant matrix and a vector can be obtained by multiplying only the first row of the circulant matrix with shifts of the vector.

The organization of this chapter is as follows: in Section 3.2 we recall the main tools of code-based cryptography, then we explain in Section 3.3 how to decrease the size of the public key by using the QC-MDPC version of McEliece cryptosystem and double circulant matrices. In Section 3.4, we consider the security of this new construction, while Section 3.5 compares the performances of the new provably secure group signature with the previous one. Section 3.6 is devoted for conclusion.

3.2 Preliminaries

In this section, we provide the notations that will be used all along this work and we give background in code-based cryptography.

3.2.1 Notations

Let λ denotes the security parameter and $\text{negl}(\lambda)$ denotes a negligible function in λ . We denote by $a \xleftarrow{\$} S$ if a is chosen uniformly at random from the finite set S . The symmetric group of all permutations of k elements is denoted by S_k . We use bold capital letters, (e.g., $\mathbf{A}$), to denote matrices, and bold lowercase letters, (e.g., $\mathbf{x}$), to denote row vectors. We use $\mathbf{x}^\top$ to denote the transpose of $\mathbf{x}$ and $\text{wt}(\mathbf{x})$ to denote the Hamming weight of $\mathbf{x}$. We denote by $B(m, \omega)$ the set of all vectors $\mathbf{x} \in \mathbb{F}_2^m$ ($\mathbf{x}$ is a vector of m bits) of hamming weight ω : $\text{wt}(\mathbf{x}) = \omega$. Throughout the chapter, we define a function $I2B$ which takes a non-negative integer a as an input, and outputs the binary representation $(a_0, \dots, a_{\ell-1}) \in \{0, 1\}^\ell$ of a such that $a = \sum_{i=0}^{\ell-1} a_i 2^{\ell-1-i}$, and a function $B2I$ which takes as an input the binary representation $(a_0, \dots, a_{\ell-1}) \in \{0, 1\}^\ell$ of a , and outputs a . All logarithms are of base 2. By 1^λ we denote the string of N ones. We denote by $\oplus$ the bitwise XOR operator.

3.2.2 Code-based Cryptography Background

Now we give some necessary notions in code-based cryptography for the well understanding of our work.

The Randomized QC-MDPC McEliece Encryption Scheme.

We derive from the QC-MDPC McEliece encryption scheme [105] a randomized variant as it is suggested in [109] for the original version [103] with Goppa codes, where a uniformly random vector is concatenated to the plaintext. We describe the scheme as follows:

- Let $\mathcal{S}(1^\lambda)$ be the setup algorithm, it selects parameters n, q, t and w which are chosen based on λ for a binary t -error-correcting (n, q, w) -QC-MDPC code. Choose integers q_1, q_2 such that $n - q = q_1 + q_2$. Set the plaintext space as $\mathbb{F}_2^{q_2}$.

- Let $\mathcal{K}(n, q, w, t)$ be the algorithm that generate the keys, it performs in two steps:
 1. Generate a parity-check matrix $\mathbf{H} \in \mathbb{F}_2^{q \times n}$ of a t -error-correcting (n, q, w) -QC-MDPC code.
 2. Generate its corresponding generator matrix $\mathbf{G} \in \mathbb{F}_2^{(n-q) \times n}$ in row reduced echelon form.

Output encrypting key $p_{kME} = \mathbf{G}$ and decrypting key $s_{kME} = \mathbf{H}$.

- Let $\mathcal{E}(p_{kME}, \mathbf{m})$ be the encryption algorithm, it encrypts a message $\mathbf{m} \in \mathbb{F}_2^{q_2}$, sample $\mathbf{u} \xleftarrow{\$} \mathbb{F}_2^{q_1}$ and $\mathbf{e} \xleftarrow{\$} B(n, t)$, then output the ciphertext $\mathbf{c} = (\mathbf{u} \parallel \mathbf{m}) \cdot \mathbf{G} \oplus \mathbf{e} \in \mathbb{F}_2^n$.
- Let $\mathcal{D}(s_{kME}, \mathbf{c})$ be the decryption algorithm and let $\mathcal{R}_{\mathbf{H}}$ be a t -error correcting Low-Density Parity Check (LDPC) decoding algorithm equipped with the knowledge of $\mathbf{H}$. To decrypt $\mathbf{c} \in \mathbb{F}_2^n$:
 1. compute $\mathbf{y} = \mathcal{R}_{\mathbf{H}}(\mathbf{c})$
 2. get the plaintext $\mathbf{m}$ as follow: extract $(\mathbf{u} \parallel \mathbf{m})$ from the first $(n - q)$ positions of $\mathbf{y}$.

Definition 17 (The Decisional McEliece problem). The $DMcE(n, k, t)$ problem is as follows: given a matrix $\mathbf{G} \in \mathbb{F}_2^{k \times n}$, distinguish whether $\mathbf{G}$ is a uniformly random matrix over $\mathbb{F}_2^{k \times n}$ or it is generated by algorithm $\mathcal{K}(n, q, w, t)$ described above.

When n, q, t are chosen based on λ and $k = n - q$, we say that the $DMcE(n, k, t)$ problem is hard, if the success probability of any probabilistic polynomial time (PPT) distinguisher is at most $\frac{1}{2} + \text{negl}(\lambda)$.

Definition 18 (The Decisional Learning Parity with (fixed-weight) Noise problem). The $DLPN(k, n, B(n, t))$ problem is as follows: given a pair $(\mathbf{A}, \mathbf{v}) \in \mathbb{F}_2^{k \times n} \times \mathbb{F}_2^n$, distinguish whether $(\mathbf{A}, \mathbf{v})$ is a uniformly random pair over $\mathbb{F}_2^{k \times n} \times \mathbb{F}_2^n$ or it is obtained by choosing $\mathbf{A} \xleftarrow{\$} \mathbb{F}_2^{k \times n}$, $\mathbf{u} \xleftarrow{\$} \mathbb{F}_2^k$, $\mathbf{e} \xleftarrow{\$} B(n, t)$ and outputting $(\mathbf{A}, \mathbf{u} \cdot \mathbf{A} \oplus \mathbf{e})$. When n, k, t are chosen based on λ , we say that the $DLPN(k, n, B(n, t))$ problem is hard, if the success probability of any PPT distinguisher is at most $\frac{1}{2} + \text{negl}(\lambda)$.

Remark 3. The Proposition 3 shows that $DMcE$ and $DLPN$ problems from (n, q, w) -QC-MDPC code are both very hard as the random case of these problems.

In the standard model assuming the hardness of the $DMcE(n, k, t)$ problem and the $DLPN(q_1, n, B(n, t))$ problem described in [57, 109] and assuming the Remark 3, the scheme described above is CPA-Secure.

The Underlying Zero-Knowledge Argument System.

We recall the Underlying Zero Knowledge Argument System used by Ezerman and al in [58] which is based on Stern Zero Knowledge Protocol [124] using random double circulant matrix instead of random matrix .

Let $n, q, t, w, p, \omega, \ell$ be positive integers, and $N = 2^\ell$. In this protocol, the public input consists of matrices $\mathbf{G} \in \mathbb{F}_2^{(n-q) \times n}$, $\mathbf{H} \in \mathbb{F}_2^{p \times 2p}$; N syndromes $\mathbf{y}_0, \dots, \mathbf{y}_{N-1} \in \mathbb{F}_2^p$; and a vector $\mathbf{c} \in \mathbb{F}_2^n$. It allows prover $\mathcal{P}$ to simultaneously convince verifier $\mathcal{V}$ in Zero Knowledge that $\mathcal{P}$ possesses a vector $\mathbf{s} \in B(2p, \omega)$ corresponding to certain syndrome $\mathbf{y}_j \in \{\mathbf{y}_0, \dots, \mathbf{y}_{N-1}\}$ with hidden index j , and that $\mathbf{c}$ is a correct encryption of $I2B(j)$ via the randomized QC-MDPC McEliece encryption. Specifically, the secret witness of $\mathcal{P}$ is a tuple $(j, \mathbf{s}, \mathbf{u}, \mathbf{e}) \in [0, N-1] \times \mathbb{F}_2^{2p} \times \mathbb{F}_2^{n-q-\ell} \times \mathbb{F}_2^n$ such that:

$\mathbf{H} \cdot \mathbf{s}^\top = \mathbf{y}_j^\top$ and $(\mathbf{u} \parallel I2B(j)) \cdot \mathbf{G} \oplus \mathbf{e} = \mathbf{c}$ where $\mathbf{s} \in B(2p, \omega)$ and $\mathbf{e} \in B(n, t)$.

3.3 Our Proposed Group Signature Scheme

In this section, we present our improved group signature scheme of [58] which is a the static group signature (see Section 2.3.1). The improvement concern the *KeyGen* algorithm which is responsible to generate group public key g_{pk} , group manager secret key g_{msk} and group secret key g_{sk} . We aim to reduce the group signature public key's length, for that, we proceed in two steps: first we use a generator matrix $\mathbf{G}$ of QC-MDPC code, as described in [105], to run the randomised QC-MDPC McEliece cryptosystem variant in order to have a reduced length of public key for the same security level achieved using Goppa codes. Second, we act at the choice of the matrix $\mathbf{H}$ used in the Zero Knowledge interactive protocol by using random double circulant matrix than a random one as presented in [65]. The algorithms *Sign*, *Verify* and *Open* remain unchanged.

Our scheme is as follows:

KeyGen $(1^\lambda, 1^N)$: takes as input a security parameter λ and the number of group users $N = 2^\ell$. Outputs: the group public key g_{pk} , group manager secret key g_{msk} and the group secret keys

g_{sk} .

Then, we select:

- Parameters: $n = n(\lambda)$, $q = q(\lambda)$, $w = w(t)$ for the (n, q, w) -QC-MDPC code.
- Parameters: $p = p(\lambda)$, $\omega = \omega(\lambda)$ for the syndrome decoding problem.
- A collision-resistant hash function

$\mathcal{H} : \{0, 1\}^* \rightarrow \{1, 2, 3\}^k$ (where $k = \omega(\log(\lambda))$) used in the Fiat Shamir paradigm [62].

Let $\mathbf{G}$ be the public key of QC-MDPC variant of McEliece cryptosystem, $\mathbf{G} \in \mathbb{F}_2^{(n-q) \times n}$ is the generator matrix of a (n, q, w) QC-MDPC code. As described in Remark 2.1, to generate the matrix $\mathbf{G}$ we need only $(n - q)$ bits. Unlike the original scheme where $\mathbf{G} \in \mathbb{F}_2^{(n-q) \times n}$ (of length $(n - q) \times n$) is a generator matrix of (n, k, t) Goppa code.

Let $\mathbf{H} = [\mathbf{I}_p \mid \mathbf{A}] \in \mathbb{F}_2^{p \times 2p}$ be a random double circulant matrix, where $\mathbf{I}_p$ is the identity matrix and $\mathbf{A}$ is a $p \times p$ circulant matrix of length p . Therefore, we need only p bits (corresponding to the first row of $\mathbf{A}$) to generate $\mathbf{H}$ which is not the case in the original scheme where $\mathbf{H}$ is chosen uniformly random. The *KeyGen* algorithm proceeds as follows:

1. Run $\mathcal{K}(n, q, w, t)$ to obtain a key pair $(p_{kME} = \mathbf{G} \in \mathbb{F}_2^{(n-q) \times n}; s_{kME})$ for the QC-MDPC variant of McEliece encryption scheme.
2. Choose a random double circulant matrix
 $\mathbf{H} = [\mathbf{I}_p \mid \mathbf{A}]$.
3. For each $j \in [0, N - 1]$, pick $\mathbf{s}_j \xleftarrow{\$} B(2p, \omega)$, and let $\mathbf{y}_j \in \mathbb{F}_2^p$ be its syndrome, i.e., $\mathbf{y}_j^\top = \mathbf{H} \cdot \mathbf{s}_j^\top$.
4. The keys are as follows:
 $g_{pk} = (\mathbf{G}, \mathbf{H}, \mathbf{y}_0, \dots, \mathbf{y}_{N-1})$, $g_{msk} = s_{kME}$, $g_{sk} = (\mathbf{s}_0, \dots, \mathbf{s}_{N-1})$.

Sign $(g_{sk}[j], M)$: the user j sign a message $M \in \{0, 1\}^*$ under g_{pk} using his secret key $s = g_{sk}[j]$ by following these steps:

1. Encrypt the binary representation of j , i.e., vector $I2B(j) \in \mathbb{F}_2^\ell$ using the QC-MDPC randomised McEliece public key $\mathbf{G}$. The cipher text is

$$\mathbf{c} = (\mathbf{u} \parallel I2B(j)) \cdot \mathbf{G} \oplus \mathbf{e} \in \mathbb{F}_2^n$$

where $\mathbf{u} \xleftarrow{\$} \mathbb{F}_2^{n-q-\ell}$ and $\mathbf{e} \xleftarrow{\$} B(n, t)$

2. Generate the signature Π by running the Zero Knowledge variant of Stern in order to prove the possession of $\mathbf{s} \xleftarrow{\$} B(2p, \omega)$ corresponding to a certain syndrome $\mathbf{y}_j \in \{\mathbf{y}_0, \dots, \mathbf{y}_{N-1}\}$ with hidden index j , and that $\mathbf{c}$ is a correct QC-MDPC McEliece encryption of $I2B(j)$. This is done by employing the Underlying Interactive Zero Knowledge Protocol with public input $(\mathbf{G}, \mathbf{H}, \mathbf{y}_0, \dots, \mathbf{y}_{N-1}, \mathbf{c})$ and prover's witness $(j, \mathbf{s}, \mathbf{u}, \mathbf{e})$ that satisfies:

$$\begin{cases} \mathbf{H} \cdot \mathbf{s}^\top = \mathbf{y}_j \wedge \mathbf{s} \in B(2p, \omega) \\ (\mathbf{u} \parallel I2B(j)) \cdot \mathbf{G} \oplus \mathbf{e} = \mathbf{c} \wedge \mathbf{e} \in B(n, t) \end{cases} \quad (3.1)$$

3. The protocol is repeated $k = \omega(\log(\lambda))$ times to achieve negligible soundness error, and then apply the Fiat-schamir heuristic [62] to transform it in a signature.

Let Π be a non-interactive zero-knowledge arguments of knowledge

$$\Pi = (CMT^{(1)}, \dots, CMT^{(k)}; (Ch^{(1)}, \dots, Ch^{(k)});$$

$RSP^{(1)}, \dots, RSP^{(k)})$ where CMT^i and RSP^i are respectively the commitment and the response for the i^{th} iteration in the interactive protocol

and

$$(Ch^{(1)}, \dots, Ch^{(k)}) = \mathcal{H}(M; CMT^{(1)}, \dots, CMT^{(k)}; g_{pk}, \mathbf{c}) \in \{1, 2, 3\}^k$$

.

4. The signature is $\Sigma = (\mathbf{c}, \Pi)$.

Verify (g_{pk}, M, Σ) : since $\Sigma = (\mathbf{c}, \Pi)$ and

$$\Pi = (CMT^{(1)}, \dots, CMT^{(k)}; (Ch^{(1)}, \dots, Ch^{(k)});$$

$RSP^{(1)}, \dots, RSP^{(k)})$, the verification occur in two steps:

1. If

$$(Ch^{(1)}, \dots, Ch^{(k)}) \neq \mathcal{H}(M; CMT^{(1)}, \dots, CMT^{(k)}; g_{pk}, \mathbf{c}), \text{ then return 0.}$$

2. For $i = 1$ to k , run the verification step of the interactive protocol in Section 2 with public input $(\mathbf{G}, \mathbf{H}, \mathbf{y}_0, \dots, \mathbf{y}_{N-1}, \mathbf{c})$ to check the validity of $RSP^{(i)}$ with respect to $CMT^{(i)}$ and $Ch^{(i)}$. If any of the verification conditions does not hold, then return 0 else return 1.

Open (g_{msk}, M, Σ) : Parse Σ as $(\mathbf{c}, \Pi)$ and run $\mathcal{D}(g_{msk}, \mathbf{c})$ to decrypt $\mathbf{c}$. If decryption fails, then return False. If decryption outputs $\mathbf{g} \in \mathbb{F}_2^\ell$ then return $j = B2I(\mathbf{g}) \in [0, N-1]$.

Theorem 2. *The group signature scheme is correct, CPA anonymous, traceable and the public key has size of $((n - q) + (N + 1)p)$ bits.*

A sketch of proof of this theorem is in Section 4.

3.4 Security Analysis

Our code-based group signature scheme satisfies all the security properties listed in Subsection 2.3.

It is clear that the size of the public key is

$((n - q) + (N + 1)p)$ bits, since $\mathbf{G}$ and $\mathbf{H}$ have size of $(n - q)$ bits and p bits respectively and for all $j \in [0, N - 1]$ $\mathbf{y}_j$ is of length p bits.

- **Correctness**

Let $j \in [0, N - 1]$ be a honest user. Consequently, there exists a tuple $(j, \mathbf{s}, \mathbf{u}, \mathbf{e})$ that satisfy Equation 3.1. Π is a valid signature thanks to the perfect completeness of the underlying Zero Knowledge Protocol, then the algorithm $\text{Verify}(g_{pk}, M, \Sigma)$ always outputs 1 for all $M \in \{0, 1\}^*$. On the other hand, let $\mathbf{c}$ be a ciphertext of the form $\mathbf{c} = (\mathbf{u} \parallel I2B(j)) \cdot \mathbf{G} \oplus \mathbf{e}$. If we run the algorithm $\mathcal{D}(g_{msk}, \mathbf{c})$, the output will be $I2B(j)$ by the correctness of the randomised QC-MDPC variant of McEliece. Finally, *Verify* and *Open* are correct which implies the correctness of the signature scheme.

- **Anonymity**

Since we use a randomized version of QC-MDPC McEliece and given the hardness of $DMcE(n, k, t)$ problem and the $DLPN(q_1, n, B(n, t))$ probleme in the QC-MDPC case (Remark 3), the CPA anonymity proof will be the same as [58].

- **Traceability**

We consider $\mathcal{A}$ a PPT adversary against the traceability of the scheme with success probability equal to ε . We construct a PPT adversary $\mathcal{F}$ that is able to solve a $SD(2p, p, \omega)$ problem with success probability polynomially related to ε . Algorithm $\mathcal{F}$ receives a

challenge $SD(2p, p, \omega)$ instance $(\hat{\mathbf{H}}, \hat{\mathbf{y}}) \in \mathbb{F}_2^{p \times 2p} \times \mathbb{F}_2^p$ with $\hat{\mathbf{H}}$ a random double circulant matrix.

The goal of $\mathcal{F}$ is to find a vector $s \in B(2p, \omega)$ such that $\hat{\mathbf{H}}\mathbf{s}^T = \hat{\mathbf{y}}^T$. It then proceeds as follows:

1. Pick a guess $j^* \in [0, N-1]$ and set $\mathbf{y}_{j^*} = \hat{\mathbf{y}}$.
2. Set $\mathbf{H} = \hat{\mathbf{H}}$. For each $j \in [0, N-1]$ such that $j \neq j^*$, sample $\mathbf{s}_j \xleftarrow{\$} B(2p, \omega)$ and set $\mathbf{y}_j \in \mathbb{F}_2^p$ be its syndrome, i.e., $\mathbf{y}_j^T = \mathbf{H}\mathbf{s}_j^T$.
3. Run $\mathcal{K}(n, q, w, t)$ to obtain a key pair $(p_{kME} = \mathbf{G} \in \mathbb{F}_2^{(n-q) \times n}, s_{kME})$.
4. Sent $g_{pk} = (\mathbf{G}, \mathbf{H}, \mathbf{y}_0, \dots, \mathbf{y}_{N-1})$, $g_{msk} = s_{kME}$ to $\mathcal{A}$.

The forger $\mathcal{F}$ then initializes a set $CU = \emptyset$ and handles the queries from $\mathcal{A}$ as follows:

- Queries to the random oracle H are handled by consistently returning uniformly random values in $\{1, 2, 3\}^k$
- $O^{Corrupt(j)}$, for any $j \in [0, N-1]$: if $j \neq j^*$, then $\mathcal{F}$ sets $CU = CU \cup j$ and gives s_j to $\mathcal{A}$; If $j = j^*$, then $\mathcal{F}$ aborts.
- $O^{Sign(j, M)}$, for any $j \in [0, N-1]$ and any message M : if $j \neq j^*$, then $\mathcal{F}$ honestly computes a signature since it has s_j ; If $j = j^*$, then $\mathcal{F}$ returns a simulated signature Σ^* .

The adversary $\mathcal{A}$ give a forged signature Σ^* on a message M^* . The requirements of the traceability experiment implies that:

$$Verify(g_{pk}, M^*, \Sigma^*) = 1 \text{ and}$$

$$Open(g_{msk}, M^*, \Sigma^*) = j^*.$$

By applying the same technics used in the traceability proof [58], $\mathcal{F}$ is able to solve the $SD(2p, p, \omega)$ with non negligible probability which contradicted the hardness of $SD(2p, p, \omega)$ problem.

3.5 Results

In Table 3.1, we compare our results with the original scheme one [58] for different size of group users and for 80 bits security level.

We use a (n, q, w, t) QC-MDPC code of parameters $n = 9602$, $q = 4801$, $w = 90$, $t = 84$ the size of the matrix $\mathbf{G}$ is $n - q = 4801$ bits (parameters for 80 bits security). It is shown in [105] that for this parameters a key distinguishing attack have a cost of $2^{80.70}$ operations and a Key recovery attack have a cost of $2^{81.04}$ operations.

We also use a double circulant matrix $\mathbf{H}$ of size $p = 350$ bits to achieve the same security level (80 bits) as it was detailed in [65].

Table 3.1: Public keys in the original scheme case and in our new scheme case

N	old key size	Our key size	factor reduction
$N=2^4$	625 KB	1.34 KB	466
$N=2^8$	642 KB	11.84 KB	54
$N=2^{12}$	906 KB	179.84 KB	5
$N=2^{16}$	5.13 MB	2.86 MB	1.79
$N=2^{20}$	72.8 MB	45.87 MB	1.58
$N=2^{24}$	1.16 GB	0.73 GB	1.58
$N=2^{28}$	18.45 GB	11.74 GB	1.57

We notice that our results, for the public key size, are better than ones obtained in the original scheme especially for groups with small size. For instance, when we have a group of $N = 2^4$ the size of our public key is 466 times shorter than the previous scheme. However when $N \geq 2^{16}$, the reduction factor tends to 1.57.

3.6 Conclusion

In this chapter, we have proposed a provably secure code based group signature scheme with reduced public key length, the public keys can be 466 times shorter than the original scheme, typically for a group of 16 users when the public key is 1.34 kB, while the size is 625 kB in the original scheme. The proposed scheme also satisfies the correctness, CPA anonymity and traceability properties which are the security requirements for a static group signature scheme. In a future work, we will try to reduce the signature size. We can also construct a scheme achieving CCA-anonymity and supporting revocation mechanism.

A New Dynamic Code-Based Group Signature Scheme

Contents

4.1	Introduction	54
4.2	The proposed dynamic group signature	55
4.2.1	Setup algorithm	56
4.2.2	AGS variant protocol with collision	56
4.2.3	High level overview	59
4.2.4	Join scheme	60
4.2.5	Signature algorithm	61
4.2.6	Verification algorithm	61
4.2.7	Open algorithm	62
4.3	Security Analysis and Practical Results	63
4.3.1	Security Analysis	63
4.3.2	Practical results	67
4.4	Conclusion	68

In this chapter based on [14], we design a dynamic code-based group signature. The dynamic case in group signature is more interesting than the static one. The general idea of this scheme consists in finding a collision between two ciphertexts using two different Quasi-cyclic Moderate Density Parity-Check (QC-MDPC) matrices in McEliece cryptosystem. We use a variation of AGS Zero-Knowledge protocol to prove the possession of the secret key and then we use the Fiat Schamir transformation to turn it into a signature. The public key and signature sizes are constants and independent of group users size and are shorter than those presented in the literature for 80 bits security level. Furthermore the proposed group signature scheme presents several advantages: it is a dynamic group signature based on error correcting code assumptions which are supposed resistant to quantum computing.

4.1 Introduction

Group signature schemes have been introduced by Chaum and van Heyst [42], in order to provide revocable anonymity to a signer, who is allowed to sign on behalf of a group. In such a scheme an authority is able, in exceptional cases, to open any group signature and thus recover the actual signer. The properties of group signature schemes make them a very important cryptographic primitive, with several applications in real-life scenarios such as e-voting schemes, e-bidding, digital right management systems, controlled anonymous printing services and other domains.

For many years, several group signatures have been introduced namely the ACJT [11], which was the first provably secure coalition-resistant scheme under the Strong RSA and DDH assumptions. Boneh, Boyen and Shacham in [30] and later Camenisch and Lysyanskaya in [36] proposed very efficient group signature schemes using bilinear maps. Independently, Nguyen and Safavi-Naini presented in [108] another group signature scheme using bilinear maps. we note that all these schemes were analyzed in the random oracle model [21].

Bellare *et al.* in [20] gave formal definitions of the security properties of group signatures, they proposed the first scheme provably secure in the standard model (while totally unpractical). Independently, Kiayias and Yung [86] (and later [87]) defined a security model. Bellare *et al.* [22] extended the [20] model to the case of dynamic groups.

All the aforementioned group signature schemes are based on number theory assumptions. However, number-theoretic based cryptography will not resist to the quantum computer. Recently, the research for post-quantum group signatures is quite active as shows these publications [3, 8, 75, 89, 90, 95]. The majority of these works are based on lattice assumptions, while in code-based cryptography we denote three group signature schemes. The two first schemes presented in Asiacrypt 2015 [58] and [8] in Secrypt 2016 are based on BMW (Bellare, Micciancio and Warinschi) model [20]. The last one is based on a slight relaxation of the BSZ (Bellare, Shi and Zang) model [22] and is proposed in [3].

In this chapter, we present a new dynamic code-based group signature scheme which is an improvement of [3]. It is based on two main ideas: the first one consists in using as Zero-Knowledge argument system the AGS (Aguilar, Gaborit and Shrek) identification scheme [1], this choice implies lower signature and public key sizes. For example, to achieve a 80 bits security level, the public key and signature sizes are 2.5 Ko and 1.32 Mo respectively in our

scheme instead of 2.5 Mo and 20 Mo in [3].

We explain in the following points how we reduce the signature and public key length:

- We use two double circulant matrices (concatenation of an identity matrix and a circulant matrix) as public key. Therefore, we can generate such matrices only from the first row of the circulant matrix since the others are obtained by applying a shift.
- AGS identification scheme [1] have low communication data than all existing code based identification schemes. In addition, the cheating probability is around $\frac{1}{2}$ witch implies less number of rounds. Thus the signature size is reduced.

The second difference between our construction and that presented in [3] consists to find cipher collision instead of syndrome collision in the joining protocol. In [3], the authors generate trapdoor matrix to make syndrome collision. Notice that in Hamming metric, the one existing trapdoor matrix is CFS signature instance [48]. To generate cipher collision, we use QC-MDPC McEliece variant where in [105] the authors assume that QC-MDPC decoding algorithm is fast and simple to implement.

This chapter is organized as follows: in Section 4.2, we explain our proposed group signature scheme and we prove its security in Section 4.3 with specific parameters. We conclude in Section 4.4.

4.2 The proposed dynamic group signature

In this section, we first propose a variation of AGS identification protocol (See Section 2.2) then, we present a high level overview of our scheme. Finally, we describe precisely the operations of the different algorithms required in the proposed group signature scheme which is a dynamic group signature (See Section 2.3.2).

Let μ denotes some randomness and the symbol $\parallel$ is for concatenation. $v[r]$ denotes the r -th coordinate of the vector v and s_r denotes the r -th symbol of the string s . $\mathbb{F}_q$ denotes the finite field of cardinality q . $\mathcal{M}_{m \times n}(\mathbb{F}_q)$ denotes matrices over $\mathbb{F}_q$ of m rows and n columns. S_w^n is the set of vectors of weight w lying in $\mathbb{F}_2^n$. $\mathcal{H}$ denotes a generic random oracle. Let $h : \mathbb{F}_2^* \rightarrow \mathbb{F}_2^n$, $h' : \mathbb{F}_2^* \rightarrow \mathcal{M}_{k \times n}(\mathbb{F}_2)$ and $h'' : \mathbb{F}_2^* \rightarrow \mathbb{F}_2^\ell$ be random oracles model. For protocols, we denote by $\mathcal{P}$ and $\mathcal{V}$ respectively the prover and the verifier. We use coding theory notation, where G and

H respectively denote generator and parity check matrices of a code. Let $H \in \mathcal{M}_{k \times n}(\mathbb{F}_q)$ and $x \in \mathbb{F}_q^n$. The product $H \cdot x^T$ is called the syndrome of x and $wt(x)$ refers to the Hamming weight of x . We define 1^λ as a string of ones λ times. We denote by $a \xleftarrow{\$} S$ if a is chosen uniformly at random from the finite set S .

We define $Pr[A = a]$ as being the probability to have the event $A = a$. Let $Exp_{B,A}^{C-b}(\lambda)$ and $Adv_{B,A}^C(\lambda)$ define respectively the experiment where adversary $\mathcal{A}$ attack the properties C of the scheme B , where the objective is to output the bit b and the advantage of the adversary $\mathcal{A}$ in the experiment $Exp_{B,A}^{C-b}(\lambda)$.

We define ρ_r as a left shift rotation of r position applied to vector u as follows:

$$\begin{aligned} \rho_r: \mathbb{F}_2^n &\rightarrow \mathbb{F}_2^n \\ u &\mapsto \rho_r(u) \end{aligned}$$

4.2.1 Setup algorithm

The following algorithm aims to output global parameters to generate our code (n, k, w) , group signature public key $g_{pk} = \left(\begin{pmatrix} R \\ Q \end{pmatrix}, w \right)$, group manager secret key $g_{msk} = (H, s_{kO})$. We initialize the secret key of the opener s_{kO} to empty set.

Algorithm 4.1 Setup algorithm

Require: 1^λ

Ensure: $params, g_{pk}$

$(\lambda, k, n, w) \in \mathbb{N}^4 \leftarrow params(1^\lambda)$

$(Q, H) \in \mathcal{M}_{k \times n}(\mathbb{F}_2) \times \mathcal{M}_{(n-k) \times n}(\mathbb{F}_2) \leftarrow QC-MDPC(params)$

$R \in \mathcal{M}_{k \times n}(\mathbb{F}_2)$ a generator matrix of a $QC-MDPC(params)$ codes

$g_{pk} \leftarrow \left(\begin{pmatrix} R \\ Q \end{pmatrix}, w \right)$

$s_{kO} \leftarrow \emptyset$

$g_{msk} \leftarrow (H, s_{kO})$

4.2.2 AGS variant protocol with collision

To build our variation of AGS protocol with ciphertext collision, the idea consists in using two related instances of AGS protocol as described below.

- The matrices R , Q and H are generated as explained in the setup algorithm (Algorithm 4.1) such that R and Q are two different generator matrices of QC-MDPC code (See

Section 2.1.1) of the same parameters. Let w_1 , w_2 and w_3 be three integers that will constitute the weight of the error vectors and all of them are bellow to the capacity correction t of codes generated by R and Q .

- The user construct the first instance by choosing randomly $(e_1, m_1) \in \mathbb{F}_2^n \times \mathbb{F}_2^k$ such that $wt(e_1) = w_1$ then, he calculates $c_1 = m_1 \cdot R + e_1$. The user must also choose $(f_2, m_2) \in \mathbb{F}_2^n \times \mathbb{F}_2^k$ such that $wt(f_2) = w' < w_2$, calculate $\bar{c}_2 = m_2 \cdot Q + f_2$ and sends c_1 , $\bar{c}_2$ to the group manager (m_2 must be chosen by the user because it is a part of its secret key and in the process we use it to verify that the group manager have H).
- The group manager receives c_1 , $\bar{c}_2$ and applies the efficient decoding algorithm, of the QC-MDPC code using the secret matrix H , ψ_H to $\bar{c}_2$ ($\psi_H(\bar{c}_2)$ to get m_2). The next step consist in: the generation of $e_2 \in \mathbb{F}_2^n$ of weight w_2 , computing $c_2 = m_2 \cdot Q + e_2$ and e_3 as follows $e_3 = c_1 + c_2$. He sets the opener secret key as follows $s_{kO}[i] = (e_2, e_3, m_2)$ and sends to the user the second part of his secret key (e_2, m_2) .
- After receiving (e_2, m_2) from the manager, the user must verify if m_2 has been well calculated and sets his secret key as follow: $s_k[i] = (e_1 + e_2, (m_1 || m_2))$ with the following condition $c_1 + c_2 + e_3 = 0$.

$$\begin{aligned}
 c_1 + c_2 + e_3 = 0 &\iff e_1 + m_1 \cdot R + e_2 + m_2 \cdot Q + e_3 = 0 \\
 &\iff e_1 + e_2 + e_3 + m_1 \cdot R + m_2 \cdot Q = 0 \\
 &\iff e_1 + e_2 + e_3 + (m_1 || m_2) \cdot \begin{pmatrix} R \\ Q \end{pmatrix} = 0
 \end{aligned} \tag{4.1}$$

We note $e = e_1 + e_2 + e_3$, $m = (m_1 || m_2)$ and $G = \begin{pmatrix} R \\ Q \end{pmatrix}$

The equation (3) becomes

$$e + m \cdot G = 0$$

with $e \in \mathbb{F}_2^n$, $m \in \mathbb{F}_2^{2k}$ and $G \in \mathcal{M}_{2k \times n}(\mathbb{F}_2)$.

private key: (e, m) with e of weight $wt(e) \leq w_1 + w_2 + w_3$, of length n and $m \in \mathbb{F}_2^{2k}$.

public key: (G, w_1, w_2, w_3) with G a matrix of size $2k \times n$ and $e + m \cdot G = 0$.

Algorithm 4.2 The AGS variant

1. $\mathcal{P}$ randomly chooses $(u, v) \in \mathbb{F}_2^k \times \mathbb{F}_2^k$ and permutations (π, π') of $\{1, 2, \dots, n\}$. Then $\mathcal{P}$ sends to $\mathcal{V}$ the commitments c_1, d_1, c_2 and d_2 such that: $c_1 = h(\pi)$, $d_1 = h(\pi')$, $c_2 = h(\pi(u \cdot R))$ and $d_2 = h(\pi'(v \cdot Q))$
 $rnd \leftarrow \{\pi', \pi, u, v\}$
 2. $\mathcal{V}$ sends a value $0 \leq r \leq k-1$ (number of shifted positions) to $\mathcal{P}$.
 3. $\mathcal{P}$ build $\rho_r(e_1), \rho_r(e_2), \rho_r(e_3)$ and sends the last part of the commitment: $c_3 = h(\pi(u \cdot R + \rho_r(e_1) + v \cdot Q + \rho_r(e_2) + \rho_r(e_3)))$ and $d_3 = h(\pi'(u \cdot R + \rho_r(e_1) + v \cdot Q + \rho_r(e_2) + \rho_r(e_3)))$
 $\alpha \leftarrow \{c_1, d_1, c_2, d_2, c_3, d_3\}$
 4. $\mathcal{V}$ sends $b \in \{0, 1\}$ to $\mathcal{P}$.
 5. Two possibilities:
 - if $b = 0$: $\mathcal{P}$ reveals $\rho_r(u + \rho_r(m_1)), \rho_r(v + \rho_r(m_2)), \pi, \pi'$.
 $ans \leftarrow \{\rho_r(u + \rho_r(m_1)), \rho_r(v + \rho_r(m_2)), \pi, \pi'\}$
 - if $b = 1$: $\mathcal{P}$ reveals $\pi(u \cdot R), \pi'(v \cdot Q), \pi(\rho_r(e_1))$ and $\pi'(\rho_r(e_2)), \pi(\rho_r(e_2) + \rho_r(e_3) + v \cdot Q)$ and $\pi'(\rho_r(e_1) + \rho_r(e_3) + u \cdot R)$.
 $ans \leftarrow \{\pi(u \cdot R), \pi'(v \cdot Q), \pi(\rho_r(e_1)), \pi'(\rho_r(e_2)), \pi(\rho_r(e_2) + \rho_r(e_3) + v \cdot Q), \pi'(\rho_r(e_1) + \rho_r(e_3) + u \cdot R)\}$
 6. $\mathcal{V}$ computes $c := check(\alpha, b, ans)$ where *check* is defined by:
 - if $b = 0$: $\mathcal{V}$ verifies that c_1, d_1, c_3 and d_3 have been honestly computed.
 - if $b = 1$: $\mathcal{V}$ verifies that c_2, d_2, c_3 and d_3 have been honestly computed. and that the weight of $\pi(\rho_r(e_1))$ and $\pi'(\rho_r(e_2))$ are w_1 and w_2 respectively.
 if all checks passed
 $c \leftarrow 1$
 Else
 $c \leftarrow 0$
 7. $\mathcal{V}$ outputs *Accept* if $c = 1$ and *Reject* otherwise
 8. Additional check for $\mathcal{V}$ who knows m_2 $ac := addcheck(m_2, \alpha, \beta, ans)$:
 - If $b = 0$ $\mathcal{V}$ checks if $d_2 = h(\pi'(\rho_r(m_2)Q + (v + \rho_r(m_2))Q))$.
 – If d_2 is well calculated $ac \leftarrow 1$ else $ac \leftarrow 0$
 - else there is no additional checks to verify.
-

Remark 4. In the case where $b = 0$, c_1 and d_1 can be easily calculated and

$$\begin{aligned}
 c_3 &= h(\pi((u + \rho_r(m_1)) \cdot R + (v + \rho_r(m_2)) \cdot Q)) \\
 &= h(\pi(u \cdot R + v \cdot Q + m_1 \cdot R + m_2 \cdot Q)) \\
 &= h(\pi(u \cdot R + v \cdot Q + \rho_r(e_1) + \rho_r(e_2) + \rho_r(e_3)))
 \end{aligned}$$

By analogy, we verify d_3 . In addition, the opener can do additional checks by verifying $d_2 = h(\pi'(\rho_r(m_2)Q + (v + \rho_r(m_2))Q))$ (the opener have m_2 and Q is public).

In the case where $b = 1$, the verifier can check c_2 and d_2 by applying h on $\pi(u \cdot R)$ and $\pi'(v \cdot Q)$ respectively. We compute c_3 and d_3 as follows: $c_3 = h(\pi'(\pi(u \cdot R) + \pi(\rho_r(e_1)) + \pi(\rho_r(e_2) + \rho_r(e_3) + v \cdot Q)))$ and $d_3 = h(\pi'(v \cdot Q) + \pi'(\rho_r(e_2)) + \pi'(\rho_r(e_1) + \rho_r(e_3) + u \cdot R))$. We note that in our AGS variant protocol, we keep the same properties of Zero-Knowledge, correctness and soundness as the original AGS identification scheme and even the same cheating probability.

4.2.3 High level overview

In our signature scheme, we consider the following actors:

Group manager ($\mathcal{GM}$) represents the authority and he is able to run the setup algorithm (Algorithm 4.1) by giving as input a security parameter, to add new members to the group (by the join protocol) and to revoke the anonymity of the signers (by open algorithm).

Group members or users who sign on behalf of the group (by executing Sign algorithm).

Outsiders users who do not belong to the group but can verify a signature granted the group public key g_{pk} (by running Algorithm 4.5).

The signature scheme occur in three steps: firstly, the user $\mathcal{U}$ asks the manager $\mathcal{GM}$ for joining the group and then he gets his secret signing key; secondly, the user sign on behalf of the group using his secret key and finally, the group manager $\mathcal{GM}$ is able to recover the signer's identity.

First step: the user $\mathcal{U}$ chooses randomly $e_1 \in \mathbb{F}_2^n$ of weight w_1 , $m_1 \in \mathbb{F}_2^k$, $m_2 \in \mathbb{F}_2^k$ and $f_2 \in \mathbb{F}_2^n$ such that $wt(f_2) < w_2$. He compute $c_1 = m_1 \cdot R + e_1$ and $\bar{c}_2 = m_2 \cdot Q + f_2$. The user $\mathcal{U}$ sends c_1 and $\bar{c}_2$ to the $\mathcal{GM}$ who generate $e_2 \in \mathbb{F}_2^n$ of weight w_2 , $m_2 \in \mathbb{F}_2^k$ and $e_3 \in \mathbb{F}_2^n$ such that $c_1 = m_1 \cdot R + e_1 = m_2 \cdot Q + e_2 + e_3$ (the collision as explained in Algorithm 4.3), the user's secret key is $(e_1 + e_2, m_1 \| m_2)$. The $\mathcal{GM}$ has access only to (e_2, e_3, m_2) , while (e_1, m_1) is known only by the user himself.

Second step: every group user possesses a secret key verifying

$$(e_1 + e_2 + e_3) + (m_1 \| m_2) \cdot \begin{pmatrix} R \\ Q \end{pmatrix} = 0$$

The possession of such secret key will be proved in Zero-Knowledge way using the AGS variant identification protocol.

Third Step: the open algorithm is running by the group manager $\mathcal{GM}$ using the second part of the user's secret key.

4.2.4 Join scheme

The Join protocol (Algorithm 4.3) is an interactive protocol between the group manager $\mathcal{GM}$ and the user $\mathcal{U}_i$. Firstly the user randomly chooses two pair $(e_1^{(i)}, m_1^{(i)}) \in \mathbb{F}_2^n \times \mathbb{F}_2^k$ and $(f_2^{(i)}, m_2^{(i)}) \in \mathbb{F}_2^n \times \mathbb{F}_2^k$ such that $wt(f_2^{(i)}) < w_2$ and $wt(e_1^{(i)}) = w_1$. He sends $c_1^{(i)} = m_1^{(i)} \cdot R + e_1^{(i)}$ and $\bar{c}_2^{(i)} = m_2^{(i)} \cdot Q + f_2^{(i)}$ to $\mathcal{GM}$ where $R, Q \in \mathcal{M}_{k \times n}(\mathbb{F}_2)$ are two generators matrix of two QC-MDPC codes of same parameters. Secondly the $\mathcal{GM}$ uses $H \in \mathcal{M}_{(n-k) \times n}(\mathbb{F}_2)$ from his g_{msk} and Q from g_{pk} to make cipher collision and compute $(e_2^{(i)}, e_3^{(i)}, m_2^{(i)}) \in S_{w_2}^n \times \mathbb{F}_2^n \times \mathbb{F}_2^k$ such that $c_1^{(i)} = m_2^{(i)} \cdot Q + e_2^{(i)} + e_3^{(i)}$ as explained in the Subsection 4.2.2.

Algorithm 4.3 Join interactive scheme

1. $\mathcal{U}_i$ chooses $(e_1^{(i)}, m_1^{(i)}) \xleftarrow{\$} \mathbb{F}_2^n \times \mathbb{F}_2^k$, $(f_2^{(i)}, m_2^{(i)}) \in \mathbb{F}_2^n \times \mathbb{F}_2^k$ such that $wt(f_2^{(i)}) < w_2$, $wt(e_1) = w_1$, then send $c_1^{(i)} = m_1^{(i)} \cdot R + e_1^{(i)}$ and $\bar{c}_2^{(i)} = m_2^{(i)} \cdot Q + f_2^{(i)}$ to $\mathcal{GM}$
 2. $\mathcal{GM}$ computes :
 - $m_2^{(i)} \cdot Q = \psi_H(\bar{c}_2^{(i)})$
 - retrieve $m_2^{(i)}$ from the first k positions of $m_2^{(i)} \cdot Q$
 - generate $e_2^{(i)} \xleftarrow{\$} S_{w_2}^n$
 - compute $c_2^{(i)} = m_2^{(i)} \cdot Q + e_2^{(i)}$ and $e_3^{(i)} = c_1^{(i)} + c_2^{(i)}$

if $\exists i_0 : s_{kO}[i_0] = (e_2^{(i)}, e_3^{(i)}, m_2^{(i)})$ return false
 else $s_{kO}[i] = (e_2^{(i)}, e_3^{(i)}, m_2^{(i)})$ and send $e_2^{(i)}, m_2^{(i)}$ to $\mathcal{U}_i$
 3. if the computed $m_2^{(i)}$ is different to the received $m_2^{(i)}$ return false
 else $s_k[i] = (e_1^{(i)} + e_2^{(i)}, m_1^{(i)} \| m_2^{(i)})$
-

4.2.5 Signature algorithm

This algorithm outputs a signature σ of a message m valid under g_{pk} using secret key $s_k[i]$ of user $\mathcal{U}_i$. The signature is obtained by applying the Fiat-Shamir transformation to our AGS Zero Knowledge identification variant. We repeat the variant l_λ times until achieving cheating probability close to 0 corresponding to λ security level. To sign a message m (Algorithm 4.4), a group member $\mathcal{U}_i$ produces a transcript $Tr = (\alpha, \beta, ans)$ of the AGS variant protocol (Algorithm 4.2) executed on public key g_{pk} and secret key $s_k[i]$ simulating the interaction through the use of a random oracle h'' . In the round r , $rnd[r]$ are the random elements generated by $gen(\mu)$ ($gen(\mu)$ is a function that generate random elements in the AGS variant protocol). To run the AGS variant, $\alpha[r]$ are the commitment calculated as described in Algorithm 4.2 the value of $\beta \in \{0, 1\}^l$ depends on the random oracle h'' (we choose h'' such that $l \geq l_\lambda$) and $ans[r]$ are responses obtained in the AGS variant protocol.

Algorithm 4.4 $\text{Sign}(g_{pk}, s_k[i], m, l_\lambda, \mu)$

$\alpha \leftarrow \emptyset, rnd \leftarrow \emptyset, ans \leftarrow \emptyset$ and $r \leftarrow 0$

1. **While** ($r < l_\lambda$)
 $rnd[r] \leftarrow gen(\mu)$
 $\alpha[r] \leftarrow com(g_{pk}, s_k[i], rnd[r])$
 $r \leftarrow r + 1$
 2. $\beta = h''(m, \alpha)$
 3. $r = 0$
 4. **While** $r < l_\lambda$
 $ans[r] \leftarrow resp(s_k[i], \beta_r, rnd[r])$
 $r \leftarrow r + 1$
 5. Output (m, σ) where $\sigma = (\alpha, ans)$
-

4.2.6 Verification algorithm

To verify a signature σ of a message m generate by Algorithm 4.5, the verifier should firstly split the signature as $\sigma = (\alpha, ans)$. Then he generates β from m and α using the random oracle h'' to verify the integrity of the message m and the membership of the user. Thereafter, we check the responses $ans[r]$ corresponding to $\alpha[r]$ and β_r for each iteration. If all checks passed, the signature is valid if not the verification fails.

Algorithm 4.5 $\text{Verif}(g_{pk}, m, l_\lambda, \sigma)$

1. Split $\sigma = (\alpha, ans)$
 $\beta = h''(m, \alpha)$
 2. $r \leftarrow 0$
While ($r < l_\lambda$)
 $c = \text{check}(\alpha[r], \beta_r, ans[r])$
if ($c == 0$)
 Output 0
 $r \leftarrow r + 1$
 3. Output 1
-

4.2.7 Open algorithm

We note that the manager secret key s_{kO} consists on a part of users secret key. We have for all users i , $s_{kO}[i] = (e_2^{(i)}, e_3^{(i)}, m_2^{(i)})$. We note also that a group manager can check more commitments than a classical verifier. Consequently $\mathcal{GM}$ may check additional requirements at Step 6 of our AGS variant protocol (Algorithm 4.2).

In the case $b = 0$, $\mathcal{GM}$ can check

$$d_2 = h(\sigma'(\rho_r(m_2)Q + (v + \rho_r(m_2))Q))$$

in addition to c_1, d_1, c_3, d_3 as described in Remark 4.

We define the function $\text{addcheck}(s_{kO}[i], \alpha[r], \beta_r, ans[r])$ which is similar to the function $\text{check}(\alpha[r], \beta_r, ans[r])$ but checks additional requirements as described above, we note that only $\mathcal{GM}$ can run the function addcheck using s_{kO} .

Given a signature $\sigma = (\alpha, ans)$ on a message m , the group manager (who execute the open algorithm 4.6) acts like in algorithm *Verify* but now with additional checks. If there exists a user i such as all iterations on r are successful, the signer can only be i .

Moreover, $\text{addcheck}(s_{kO}[i], \alpha[r], \beta_r, ans[r])$ outputs 1 with a probability around to $\frac{1}{2}$ when i is not the real signer of the message m . Because only the case $\beta_r = 1$ who can pass (their is no additional check to do). We recall that the cheating probability in our AGS variant protocol is asymptotically equal to $\frac{1}{2}$, thus l_λ rounds required for algorithms *Sign* and *Verify* ensure an opening probability close to 0.

Algorithm 4.6 Open algorithm

Require: $s_{kO}, g_{pk}, m, \sigma$
Ensure: i : the signer identity
 Split $\sigma = (\alpha, ans)$
 $\beta = h''(m, \alpha)$
 $i \leftarrow 1$
while $i < \text{number of Users}$ **do**
 $r \leftarrow 0$
 while $r < \ell_\lambda$ **do**
 $c \leftarrow \text{addCheck}(s_{kO}[i], \alpha[r], \beta_r, ans[r])$
 if $c = 0$ **then**
 $i \leftarrow i + 1$
 $r \leftarrow 0$
 else if $c = 1$ and $r = \ell_\lambda - 1$ **then**
 return i
 else
 $i \leftarrow i + 1$
 end if
 end while
end while
return *False*

4.3 Security Analysis and Practical Results

4.3.1 Security Analysis

In this part we study the security requirements as it was defined in Algorithm 2.3 and Algorithm 2.4 of our signature scheme. We start with anonymity properties and finish with Soundness properties. Our methodology is similar to the one used in the extended version of [3].

Anonymity: We now study the anonymity property.

Theorem 3. *If there exists an adversary $\mathcal{A}$ that can break the anonymity property of the scheme, then there exists an adversary $\mathcal{B}$ that can break the Zero-Knowledge property of our AGS variant.*

Proof. Let $\mathcal{A}$ be an adversary against the anonymity of our scheme with advantage ε . We will prove that ε is negligible using the Zero-Knowledge property of our AGS variant. We consider the following sequence of games : $\mathcal{G}_*$, $\mathcal{G}_0$ and $\mathcal{G}_1$.

- $\mathcal{G}_*$: the challenger run $Setup(1^\lambda)$ and gives g_{pk} to $\mathcal{A}$ who has also access to oracles joinP, corrupt, sign and open. This game is the same as the real anonymity experiment (Algorithm 2.4), $\mathcal{A}$ outputs (i_0, i_1, m) , but the challenger generates a simulated signature σ^* by programming the random oracle $\mathcal{H}_\lambda$ accordingly.
- $\mathcal{G}_0$: in this game the challenger and the adversary $\mathcal{A}$ has the same way and access to oracles as in game $\mathcal{G}_*$. Now the challenger and the adversary execute the real anonymity experiment (Algorithm 2.4) for $b = 0$. $\mathcal{A}$ outputs (i_0, i_1, m) and the challenger outputs $\sigma_0 = \text{Sign}(g_{pk}; s_k[i_0]; m; \mu)$.
- $\mathcal{G}_1$: in this game the challenger and the adversary $\mathcal{A}$ has the same way and access to oracles as in game $\mathcal{G}_*$. Now the challenger and the adversary execute the real anonymity experiment (Algorithm 2.4) for $b = 1$. $\mathcal{A}$ output (i_0, i_1, m) and the challenger outputs $\sigma_1 = \text{Sign}(g_{pk}; s_k[i_1]; m; \mu)$.

Given that our AGS version is Zero-Knowledge, in first time, we have that, σ_* is statistically close to σ_0 and in second time, we have that, σ_* is statistically close to σ_1 . It then follows that $\text{Exp}_{GS,A}^{\text{anon}-1}(\lambda)$ and $\text{Exp}_{GS,A}^{\text{anon}-0}(\lambda)$ are indistinguishable so $\text{Adv}_{GS,A}^{\text{anon}}(\lambda) = \Pr[\text{Exp}_{GS,A}^{\text{anon}-1}(\lambda) = 1] - \Pr[\text{Exp}_{GS,A}^{\text{anon}-0}(\lambda) = 1]$ (where $\text{Exp}_{GS,A}^{\text{anon}-1}(\lambda)$ and $\text{Exp}_{GS,A}^{\text{anon}-0}(\lambda)$ are as defined in Algorithm 2) is negligible.

□

Soundness: the soundness of our group signature scheme allows us to prove the traceability and the non-frameability.

We suppose that the adversary $\mathcal{A}$ can forge a signature on user $\mathcal{U}_{j^*}$ which is considered as an uncorrupted user. The adversary $\mathcal{A}$ produces a signature

$$\sigma = (cmt; rsp) = (cmt[1], \dots, cmt[l_\lambda], rsp[1], \dots, rsp[l_\lambda])$$

such as $\text{Verif}(g_{pk}, m, \sigma) = 1$. $\mathcal{A}$ does not have any knowledge on the user secret key $s_k[j^*]$ where l_λ is the number of iteration required in the AGS variation to achieve a certain security level.

Producing such as forgery by $\mathcal{A}$ means that he have been able to successfully run l_λ iterations of AGS variation protocol without knowing a valid secret whereas the cheating probability is close to $\frac{1}{2}$. Then $\mathcal{A}$ has either broken the soundness of the AGS variation or enabled the design of a knowledge extractor reaping benefits of the forgery to produce a valid solution to the general syndrome decoding problem.

The traceability and non-frameability are two notions closely related, consequently we treat both of them simultaneously. In both cases, the adversary $\mathcal{A}$ that can produce a valid forgery σ verifying:

$$\text{Verif}(g_{pk}, m, \sigma) = 1$$

However, breaking traceability implies for $\mathcal{A}$ to produce σ such as the group manager could not trace it back to any group member. More explicitly, to attack the traceability on user $\mathcal{U}_{j^*}$, $\mathcal{A}$ should produce a forgery σ on message m such that:

$$\text{Verif}(g_{pk}, m, \sigma) = 1 \text{ and } \text{Open}(s_{kO}, g_{pk}, m, \sigma) = 0 \quad (4.2)$$

while the non-frameability requires to produce a signature that does trace back to an actual group member. The adversary produce σ verifying:

$$\text{Verif}(g_{pk}, m, \sigma) = 1 \text{ and } \text{Open}(s_{kO}, g_{pk}, m, \sigma) = j^* \quad (4.3)$$

With the evidence that $\sigma \neq \text{Sign}(g_{pk}, s_k[j^*], m; \mu)$.

Theorem 4. *If there exists an adversary $\mathcal{A}$ against the traceability (resp the non frameability) of the scheme, then we can build an adversary $\mathcal{B}$ that can either break a general decoding problem GD, or the Simulation-Soundness of the AGS variation protocol.*

Proof. Assume that there exists an adversary $\mathcal{A}$ against the traceability of our group signature with success probability equal to ϵ . We use a sequence of games to show that if $\mathcal{A}$ is efficient which means he can break the traceability property (resp the non-frameability) then it's possible for an adversary $\mathcal{B}$ to solve a difficult problem with non negligible probability related to ϵ .

$\mathcal{G}_0$: this is the original game of the traceability (non frameability) as defined in Algorithm 2.3 a (resp in Algorithm 2.3 b) when the adversary $\mathcal{B}$ runs algorithm $\text{Setup}(1^\lambda)$, he chooses a user $\mathcal{U}_{j^*}$ on which $\mathcal{A}$ will be challenged. Then $\mathcal{B}$ provides $g_{pk} = \left(\begin{pmatrix} R \\ Q \end{pmatrix}, \omega \right)$ (resp

$g_{pk} = \left(\left(\begin{pmatrix} R \\ Q \end{pmatrix}, \omega \right) \right)$ and s_{kO} to $\mathcal{A}$ which has access to the following oracles: *joinP*, *Sign*, *Open* and *corrupt* (resp *joinP*, *Sign* and *corrupt* but not *Open* since he has s_{kO}). For any query of $\mathcal{A}$, $\mathcal{B}$ responds honestly but the game aborts if $\mathcal{A}$ tries to corrupt $\mathcal{U}_{j^*}$.

At some point $\mathcal{A}$ produces a forgery (m, σ) such that for all $j \in CU$, the signature on m were never queried and the probability for $\mathcal{A}$ to have $\text{Verify}(g_{pk}, m, \sigma) = 1$ is equal to ε .

$\mathcal{G}_1$: this game is similar to the previous one $\mathcal{G}_0$ $\mathcal{A}$ still knows g_{pk} (resp g_{pk} and s_{kO}) with the same oracle access. It differ from $\mathcal{G}_0$ by the following: whenever $\mathcal{A}$ queries oracle sign on user $\mathcal{U}_{j^*}$, $\mathcal{B}$ generates a simulated valid signature. Like previously, the game aborts if $\mathcal{A}$ tries to corrupt $\mathcal{U}_{j^*}$.

At some point, $\mathcal{A}$ produces a forgery (m, σ) under the condition that for all $j \in CU$, signatures on m were never queried and signatures honestly produced on behalf of $\mathcal{U}_{j^*}$ in game $\mathcal{G}_0$ are indistinguishable from random ones produced in this game.

Consequently, for the adversary $\mathcal{A}$, games $\mathcal{G}_0$ and $\mathcal{G}_1$ are indistinguishable and the probability for $\mathcal{A}$ to have $\text{Verif}(g_{pk}, m, \sigma) = 1$ is still ε .

Under the soundness of AGS variation protocol, we now treat separately traceability (game $\mathcal{G}_1^{\text{trac}}$) and non-frameability (game $\mathcal{G}_1^{\text{nf}}$).

$\mathcal{G}_1^{\text{trac}}$: $\mathcal{A}$ has been given a forgery σ verifying (4.2). Since the AGS variation protocol is sound, $\mathcal{B}$ could thus apply a knowledge extractor algorithm on σ to generate a (e^*, m^*) passing AGS variation protocol ($m^* \cdot G + e^* = 0$). During this game $\mathcal{A}$ queried for oracle corrupt on all users except for $\mathcal{U}_{j^*}$ meaning that he may have obtained many secret keys solving the instance of general decoding $(G, \omega_1 + \omega_2 + \omega_3, 0)$. It mean that $\mathcal{A}$ has obtained the following GD instance $(G, \omega_1 + \omega_2 + \omega_3, 0, \text{usk}[i]_{i \in CU})$. If $(e^*, m^*) \notin \text{usk}[i]_{i \in CU}$, $\mathcal{A}$ has then enabled $\mathcal{B}$ to find solution to the previous GD instance with probability directly related to ε , else, $\mathcal{A}$ replays the game until $\mathcal{B}$ gets a new solution to the aforesaid GD instance.

$\mathcal{G}_1^{\text{nf}}$: in the non frameability game, $\mathcal{A}$ has a forgery σ verifying (5). Since $\mathcal{A}$ has the s_{kO} it means that he has a knowledge of a part of the user's secret key because $\text{usk}[i] = (e_1^{(i)} + e_2^{(i)}, m_1^{(i)} \| m_2^{(i)})$ and $\mathcal{A}$ has the $(e_2^{(i)}, e_3^{(i)}, m_2^{(i)})$ for all i , we have that $\mathcal{A}$ can compute $c_1^{(i)} = m_2^{(i)} \cdot Q + e_2^{(i)} + e_3^{(i)} = (m_1^{(i)} \cdot R + e_1^{(i)})$.

Using the *corrupt* oracle, $\mathcal{A}$ can learn the entire $(e_i, m_i) = (e_1^{(i)} + e_2^{(i)} + e_3^{(i)}, m_1^{(i)} \| m_2^{(i)})$ for every user (different from $\mathcal{U}_{j^*}$) he might corrupt. In fact, $\mathcal{A}$ has obtained the following unsolved GD instances $(R, c_1^{(i)}, \omega_1)_{i \in HU}$ containing the particular one $(R, c_1^{(j^*)}, \omega_1)$. Now, under the soundness of AGS variation protocol, $\mathcal{B}$, by programming the ROM, exploits σ to get

$(e^*, m^*) = (e_1^* + e_2^* + e_3^*, m_1^* \| m_2^*)$ from which he can issues signatures verifying (5) just like σ . In particular, it means that applying algorithm *Open* on signatures issued with (e^*, m^*) returns j^* . It leads to $(e_2^*, e_3^*, m_2^*) = (e_2^{(j^*)}, e_3^{(j^*)}, m_2^{(j^*)})$ and then that (e_1^*, m_1^*) is a solution to the GSD instance $(R, c_1^{(j^*)}, \omega_1)$.

At the end of game $\mathcal{G}_1$, $\mathcal{B}$ has either broken the soundness of AGS variation protocol or been able to solve a computational problem with non negligible probability related to ε . This concludes the proof. $\square$

4.3.2 Practical results

To instantiate our scheme, we propose the following parameters :

- In [76], Qian Guo *et al.* show that it's possible to successfully recovers the secret key of QC-MDPC instance for 80 bit security proposed by Misoczki *et al.* in [105]. Consequently, it's recommended to use this proposed instance of QC-MDPC for 128 bit security: $n = 19714$, $p = 9857$, $w = 142$, and $t = 134$.
- We generate R and Q two QC-MDPC matrices for the same parameters (n, p, w, t) . Generating each one of these matrices require only 9857 bits and $w = 142$ can be stocked using 8 bits which implies a public key of size $size_{g_{pk}} = (2 \times (n - p) + 8)$ -bit = 19750 bit = 2.5KB since $g_{pk} = (\begin{pmatrix} R \\ Q \end{pmatrix}, w)$.
- To achieve a $\lambda = 80$ bits security level in our AGS variation protocol, the number of rounds must be equal to $l_\lambda = 88$. We parse a signature σ as follows:

$$\sigma = (cmt[1], \dots, cmt[l_\lambda], rsp[1], \dots, rsp[l_\lambda])$$

. For each round $1 \leq i \leq l_\lambda$ $cmt[i] = (c_1, d_1, c_2, d_2, c_3, d_3)$ and we choose a hash function that returns a 160 bit then the size of each $cmt[i]$ is $size_{cmt[i]} = 6 \times 160$ bits = 960 bits.

Now we calculate the size of $rsp[i]$ such $1 \leq i \leq l_\lambda$, for this reason we calculate the size of permutations as follow:

$size_{perm} = 2 \times m \times 2^m$ -bit where $m = \frac{\lceil \log_2(2p) \rceil}{2}$ which implies a permutation of size $size_{perm} = 4096$ bits.

We distinguish 2 cases:

- The first case: where the challenge $b = 0$, the size of the response is $size_{rsp0} = 2 \times p + 2 \times size_{perm} = 27906$ bits
- The second case: where the challenge $b = 1$, the size of the response is $size_{rsp1} = 6 \times n = 118284$ bits

The size of a signature $\sigma = (cmt[1], \dots, cmt[l_\lambda], rsp[1], \dots, rsp[l_\lambda])$ is $size_\sigma = l_\lambda \times (size_{cmt[i]} + \max(size_{rsp0}, size_{rsp1})) = 10493472$ bits, which means a signature of size $1.32MB$.

Comparing our results with the first dynamic group signature based on coding theory [3] for 80 bits security level, Our schema is more efficient in terms of public key and signature sizes which are respectively equal to $2.5MB$ and $20MB$ in [3] and only $2.5KB$ and $1.32MB$ in our scheme.

Our construction is not based on trapdoor matrix, to make syndrome collision. Notice that in hamming metric, the one existing trapdoor matrix is CFS signature instance [48]. We propose an other approach in the join protocol based on QC-MDPC McEliece to make cipher collision instead of syndrome collision.

4.4 Conclusion

In this work, we have proposed a new dynamic group signature scheme based on coding theory assumptions which improve the first code based group signature presented in [3]. In this proposed scheme, group public key and signatures sizes remain independent of group length but just depending on the security level. Our technique consist in using the QC-MDPC version of McEliece cryptosystem in the join protocol and AGS identification scheme as Zero-Knowledge argument system unlike using the trapdoor matrix and Stern identification scheme as described in [3]. Even if in [76] the authors proposed a key recovery attack for 80 bits security level, our scheme remain still secure because we use parameters for 128 bits security level. We have started to develop this idea before the publication of [76]. These choices make the join protocol very fast and decreases the public key and signature length typically from $2.5MB$ in the original scheme to $2.5KB$ in our one for the public key and from $20MB$ in the original scheme to $1.32MB$ in our scheme for the signature. In future work we will try to add the revocation properties and try to turn this scheme to a List signature scheme.

An Efficient Identification Scheme Based on Rank Metric

Contents

5.1	Introduction	69
5.2	The Proposed Zero-knowledge Identification Protocol	72
5.2.1	Key Generation	74
5.2.2	Identification Protocol	74
5.3	Properties and Security of the Scheme RankId	75
5.4	Parameters and Results	81
5.5	Conclusion	83

In this chapter based on [15], using random double circulant codes, we design a rank metric version of Aguilar, Gaborit and Schrek (AGS) identification scheme which is resistant to attacks using quantum computers. We achieve optimum results in different scales comparing public key size, secret key size and communication cost with known identification schemes. Moreover, our protocol is more efficient and practical for different resource constrained devices such as smart cards or Radio Frequency Identification (RFID) tags. Furthermore, using the Fiat-Shamir paradigm, we design an efficient signature scheme.

5.1 Introduction

In 1994, assuming the existence of a quantum computer, Shor has proved in [120] that the discrete logarithm and the factorization problems can be solved in polynomial time. This paper has enabled the emergence of post-quantum cryptography. The security of the most important post-quantum primitives relies on the difficulty of solving decoding problems in specific metrics (Hamming metric for codes, Euclidean metric for lattices for example). The security of an identification scheme is based on the security of the difficult problem to which it is linked.

Identification protocols are very useful in the electronic communications area. For example, identification protocols are used in ticketing, e-passport, e-health, control access systems, etc.

The choice of code-based cryptography for identification schemes is mainly motivated by the capacity of future quantum computer. An identification scheme is said to be Zero-knowledge if it is an interactive protocol where one proves the knowledge of something, without revealing any information on this knowledge. This implies that the verifier cannot learn the secret, also he or she cannot replay the protocol and especially he or she cannot convince another person that the prover was aware of the secret.

Over the years, many Zero-knowledge identification schemes have been proposed using Hamming metric [1, 40, 123, 127]. Based on the instances of Syndrome Decoding problem, Stern presented in [123] a pioneering work that is the first code-based Zero-knowledge identification scheme whose cheating probability is $\frac{2}{3}$. In this scheme, the prover has to convince the verifier that he or she knows a word e of small weight such that $He^\top = y$ where the matrix $H \in \mathcal{M}_{(n-k) \times n}(\mathbb{F}_2)$ which is a parity check matrix of a linear binary code of length n and dimension k and the vector $y \in \mathbb{F}_2^{(n-k)}$ are public parameters. Thereafter, to decrease the cheating probability to $\frac{1}{2}$, Stern proposed in [123] an extended version of his first protocol that consists on five pass rather than three pass. This improvement impacts a reduction of the signature size. But on the other hand, it increases the communication cost. The second advanced work was presented in [127] by Veron where he proposed a different formulation of secret key using the General Decoding (GD) problem (see below). In this case the prover has to prove his knowledge of a couple of vectors (f, e) that verifies $x = fG \oplus e \in \mathbb{F}_2^{(n-k)}$ where the matrix $G \in \mathcal{M}_{n \times k}(\mathbb{F}_2)$ and x are public parameters. This new formulation decreases the communication cost but increases the key size. In [65], Gaborit and Giraut proposed to use a double circulant matrices to represent the public matrix. This proposition enables to decrease dramatically the public key size. In [40] using a q -ary codes, Cayrel *et al.* proposed another protocol with cheating probability of $\frac{1}{2}$. The last and the most efficient protocol in Hamming metric has been presented by Aguilar *et al.* in [1] which is based on the Veron's scheme and achieves a cheating probability of $\frac{1}{2}$ while preserving a low communications cost.

Chen in [44] proposed a new code-based Zero-knowledge identification scheme using another metric called rank metric. This new three pass identification scheme differs from Stern protocol [123] on the following points:

- The lack of hash function; the first version of Chen identification scheme [44] did not use hash function but matrix multiplication. This property has been very useful since the use of hash function increases the complexity of the identification scheme.

- The choice of another family of error correcting codes; Stern has used a family of error correcting codes using Hamming metric but Chen protocol [44] has used codes in rank metric. This choice can provide smaller key size than Hamming metric.

A few month after the publication of Chen's identification scheme [44], Chabaud and Stern have presented in [41] the first attack on the syndrome decoding problem in rank metric. Thus, to keep Chen's protocol safe, Chabaud and Stern [41] proposed to recalibrate the security parameters. However, the second attack presented by Gaborit *et al.* in [67] that is based on the lack of a hash function allows a passive attacker to quickly recover the user secret. To repair the Chen's protocol, Gaborit *et al.* [67] proposed a new version with hash function. This new protocol can be seen as the rank version of Stern's protocol. Recently the research in rank metric codes applied to the cryptography area is quite active with the publication of many paper like [67, 79].

In this chapter we present a five pass Zero-knowledge code-based identification scheme in metric rank with a cheating probability of $\frac{1}{2}$. Like in AGS protocol [1], our protocol uses double-circulant matrices to represent public generator matrix. Moreover the matrices used to mask user secrets in the protocol running are also circulant. This technique combined with rank metric instead of Hamming metric provide an efficient identification scheme with small key size and low communications cost. In practice, our scheme is more efficient since the public data are 276 bits rather than 350 bits in AGS protocol for 80 bits security level which implies a reduction of 21%. Moreover, for a security level of 128 bits, we get 434 bits as public data size that is better than 550 bits in the AGS protocol, which implies a gain of 21%. Also, for a security of 2^{80} , we get a signature size of 83160 bits which is better than AGS [1] case where the signature size is about 93000 bits. By applying the Fiat-Shamir paradigm [62] to a Zero-knowledge identification scheme it is easy to obtain a signature scheme.

The reminder of this chapter is organized as follows: Section 5.2 is devoted to our Zero-knowledge identification scheme. We present the security analysis of our protocol in Section 5.3 and give some practical results in Section 5.4. Then we conclude in Section 5.5.

5.2 The Proposed Zero-knowledge Identification Protocol

In this section we describe our proposed Zero-knowledge identification protocol that we call RankId. The Zero-knowledge protocols such as Shamir's Permuted-Kernel-Problem-based protocol [118], Chens scheme [45], Stern's Constrained-Linear-Equations-based scheme [125], Cayrel-Véron-ElYousfi protocol [40] and Aguilar-Gaborit-Shreck protocol [1] are all of them a five pass protocol and their security is not based on number theory. Beside having a cheating probability close to $\frac{1}{2}$, they all have one more thing in common: after a prover has sent the commitments, a verifier $\mathcal{V}$ chooses a random element in the base field that it sent to a prover $\mathcal{P}$. Then using this element combined with its secret and all masked using various techniques, the prover sends a last commitment to the verifier. In this chapter we propose to adapt these techniques to the syndrome decoding problem in rank metric. Our technique is mainly based on the well exploiting of the double circulant structure of the public matrix. We first introduce a special multiplication law that we will use in our protocol.

Let q be a power of a prime p , m an integer and let $V(n)$ be an n dimensional vector space over the finite field $\mathbb{F}_{q^m}$. We denote by $GL_m(\mathbb{F}_q)$ the group of invertible $(m \times m)$ matrices over $\mathbb{F}_q$. Let $\beta = (\beta_1, \dots, \beta_m)$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Let $\mathcal{F}_i$ be the map from $\mathbb{F}_{q^m}$ to $\mathbb{F}_q$ where $\mathcal{F}_i(x)$ is the i -th coordinate of x in the basis β . To any $v = (v_1, \dots, v_n)$ in $V(n)$ we associate the matrix $\bar{v} \in \mathcal{M}_{m,n}(\mathbb{F}_q)$ where

$$\bar{v} = (\bar{v}_{i,j})_{i=1,\dots,m; j=1,\dots,n} = \mathcal{F}_i(v_j) \quad (5.1)$$

The rank weight of a vector v noted $rank(v)$ is defined as the rank of the associated matrix $\bar{v}$. For a given basis β , we denote by Φ_β the invert of the map

$$V(n) \rightarrow \mathcal{M}_{m,n}(\mathbb{F}_q) : v \mapsto \bar{v} \quad (5.2)$$

computed in the basis β . let S be a finite set and a be an element of S we write $a \xleftarrow{\$} S$ if a is chosen uniformly at random in the finite set S . We denote by $\oplus$ the bitwise XOR operator. We define $P(A = a)$ as being the probability to have the event $A = a$. We denote by h a hash function.

Let r be an integer such that $r > 1$, to any $\alpha \in \mathbb{F}_{q^m}^*$ we associate the symmetric matrix $\tilde{\alpha}_r \in \mathcal{M}_{r,r}(\mathbb{F}_{q^m})$ such that:

$$\tilde{\alpha}_r = \begin{pmatrix} \alpha & & & \\ & \alpha^2 & & \\ & & \ddots & \\ & & & \alpha^r \end{pmatrix} \quad (5.3)$$

Definition 19 (Bullet Product "•"). *Let α be an element of $\mathbb{F}_{q^m}^*$, r be an integer such that $r > 1$ and $v = v_1 \parallel v_2$ (where $\parallel$ is the concatenation symbol) be a vector of $\mathbb{F}_{q^m}$ such that $v_1, v_2 \in V(r)$. We define the product $v \bullet \alpha$ as follow:*

$$v \bullet \alpha = v_1 \tilde{\alpha}_r \parallel v_2 \tilde{\alpha}_r \quad (5.4)$$

Proposition 5. *For any integer $r > 1$, $\alpha \in \mathbb{F}_{q^m}^*$ and v a vector of $\mathbb{F}_{q^m}$.*

$$\text{rank}(v \bullet \alpha) = \text{rank}(v) \quad (5.5)$$

Proof. Let A be an $(m \times n)$ -matrix over $\mathbb{F}_{q^m}$. In this proof we denote by $\mathcal{C}(A)$ the subspace of $\mathbb{F}_{q^m}^n$ generated by the vectors column of A . Since $v = (v_1 \parallel v_2)$ where v_1, v_2 are in $V(k)$ we have:

$$\begin{aligned} \text{rank}(v) &= \text{rank}(\bar{v}_1 \parallel \bar{v}_2) \\ &= \text{Dim}(\mathcal{C}(\bar{v}_1) + \mathcal{C}(\bar{v}_2)) \end{aligned} \quad (5.6)$$

Then we have:

$$\begin{aligned} \text{rank}(v \bullet \alpha) &= \text{rank}(v_1 \tilde{\alpha}_r \parallel v_2 \tilde{\alpha}_r) \\ &= \text{rank}(\bar{v}_1 \tilde{\alpha}_r \parallel \bar{v}_2 \tilde{\alpha}_r) \\ &= \text{Dim}(\mathcal{C}(\bar{v}_1 \tilde{\alpha}_r) + \mathcal{C}(\bar{v}_2 \tilde{\alpha}_r)) \end{aligned} \quad (5.7)$$

Because $\tilde{\alpha}_r$ is symmetric and invertible ($\alpha \neq 0$) we get the following equation:

$$\mathcal{C}(\bar{v}_1 \tilde{\alpha}_r) = \mathcal{C}(\tilde{\alpha}_r \bar{v}_1^\top) = \mathcal{C}(\bar{v}_1) \quad (5.8)$$

and

$$\mathcal{C}(\bar{v}_2 \tilde{\alpha}_r) = \mathcal{C}(\tilde{\alpha}_r \bar{v}_2^\top) = \mathcal{C}(\bar{v}_2) \quad (5.9)$$

by (5.6), (5.7), (5.8) and (5.9) we deduce that:

$$\begin{aligned} \text{rank}(v \bullet \alpha) &= \text{Dim}(\mathcal{C}(\bar{v}_1) + \mathcal{C}(\bar{v}_2)) \\ &= \text{rank}(v) \end{aligned} \tag{5.10}$$

□

5.2.1 Key Generation

Our Zero-knowledge protocol uses a public $k \times n$ (with $n = 2k$) random double circulant matrix G over $\mathbb{F}_{q^m}$. This matrix G can be considered as a generator matrix of an $[n, k]$ rank-metric code over $\mathbb{F}_{q^m}$. To get a very compact public key we can consider the matrices of type $G = (I_k, G_1)$ where G_1 is a $k \times k$ matrix over $\mathbb{F}_{q^m}$ and I_k is the $k \times k$ identity matrix. Let κ be a security level. The key generation process which is described in Algorithm 5.1 takes as input κ and outputs the public parameters (k, m, q, r) , the user public key p_k and the user secret key s_k . Let W denotes the workfactor of the best known attacks on Rank Syndrome Decoding Problem (Problem 3).

Algorithm 5.1 Key generation algorithm

Input: κ

Output: $p_k, s_k, (k, m, q, r)$

Choose k, m, q and r such that $W(k, m, q, r) \geq 2^\kappa$

$G \xleftarrow{\$} \mathcal{M}_{k, 2k}(\mathbb{F}_{q^m})$

$e \xleftarrow{\$} V(2k)$ such that $\text{rank}(e) = r$

$f \xleftarrow{\$} V(k)$

$x \leftarrow fG \oplus e$

$p_k \leftarrow (G, r, x), s_k \leftarrow (e, f)$

5.2.2 Identification Protocol

In our Zero-knowledge protocol (RankId) described in Algorithm 5.2, to prove its identity, a prover has to prove the knowledge of the secret key (e, f) by using two blinding techniques. The first one is to Xor a random vector to the secret key f and unlike the case of Hamming metric where it is common to use a permutation, the second blinding technique is to use the

" $\bullet$ " and " $*$ " products (See Definition 8) to multiply the secret e to random values. Moreover the security of our protocol relies on the hardness of the rank general decoding problem, that is on the difficulty of finding a couple (e, f) such that $x = fG \oplus e$ where x and G are public.

Algorithm 5.2 The proposed identification protocol (RankId)

1. A prover $\mathcal{P}$ randomly chooses $u \in V(k)$, $P \in Gl_{2k}(\mathbb{F}_q)$ and $Q \in Gl_m(\mathbb{F}_q)$. Then $\mathcal{P}$ sends to a verifier $\mathcal{V}$ the commitments c_1 and c_2 such that: $c_1 = h(P \parallel Q \parallel (uG_1 \oplus f))$ and $c_2 = h(Q * (uG)P)$. Where h is a hash function.
 2. A verifier $\mathcal{V}$ sends $\alpha \xleftarrow{\$} \mathbb{F}_{q^m}^*$ to $\mathcal{P}$.
 3. A prover $\mathcal{P}$ builds $c_3 = h(Q * (uG \oplus (e \bullet \alpha))P)$.
 4. A verifier $\mathcal{V}$ sends $g \in \{0, 1\}$ to $\mathcal{P}$:
 5. Two possibilities:
 - if $g = 0$: $\mathcal{P}$ reveals $(u \oplus f\tilde{\alpha}_k)$, $(uG_1 \oplus f)$ and $(P \parallel Q)$.
 - if $g = 1$: $\mathcal{P}$ reveals $Q * ((uG))P$ and $Q * (e \bullet \alpha)P$.
 6. Verification Step, two possibilities:
 - if $g = 0$: $\mathcal{V}$ verifies that c_1, c_3 have been honestly computed;
 - if $g = 1$: $\mathcal{V}$ verifies that c_2, c_3 have been honestly computed and $rank(Q * (e \bullet \alpha)P) = r$.
-

5.3 Properties and Security of the Scheme RankId

In this section we study the completeness and soundness of our identification scheme presented in Algorithm 5.2. We also show that this protocol is Zero-knowledge with cheating probability around $\frac{1}{2}$.

Completeness. We obtain the completeness of RankId described in Algorithm 5.2 by showing that if an honest prover $\mathcal{P}$ and an honest verifier $\mathcal{V}$ execute our protocol it always succeeds.

Theorem 5. *If a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ honestly execute RankId , we have for any round*

$$\Pr[\text{RankId}_{\mathcal{P},\mathcal{V}} = \text{Accept}] = 1.$$

Proof. The completeness proof is natural since a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ are supposed to be honest. We just notice that to verify c_3 in the case $g = 0$ the verifier $\mathcal{V}$ can compute

$$\begin{aligned} (u \oplus f\tilde{\alpha}_k)G \oplus x \bullet \alpha &= uG \oplus f\tilde{\alpha}_kG \oplus (fG \oplus e) \bullet \alpha \\ &= uG \oplus f\tilde{\alpha}_kG \oplus fG \bullet \alpha \oplus e \bullet \alpha \\ &= uG \oplus f\tilde{\alpha}_kG \oplus (f\tilde{\alpha}_k \parallel fG_1\tilde{\alpha}_k) \oplus e \bullet \alpha \\ &= uG \oplus f\tilde{\alpha}_kG \oplus (f\tilde{\alpha}_k \parallel f\tilde{\alpha}_kG_1) \oplus e \bullet \alpha \\ &= uG \oplus f\tilde{\alpha}_kG \oplus f\tilde{\alpha}_kG \oplus e \bullet \alpha \\ &= uG \oplus e \bullet \alpha \end{aligned}$$

And in the case $g = 1$ by applying Propositions 4 and 5 we can check that

$$\text{rank}(Q * (e \bullet \alpha)P) = r.$$

□

Zero-Knowledge. To ensure the Zero-knowledge, we use the classical idea of resettable simulation presented in [72].

Theorem 6. *The protocol defined in Algorithm 5.2 is a prover-verifier Zero-knowledge protocol in random oracle model.*

Proof. Let $\mathcal{S}$ and δ be respectively a simulator using a dishonest verifier and the number of rounds that take an honest identification process execution. Given two interactions with the prover, we assume that the dishonest verifier could construct the two following strategies: the first one $St_1(c_1, c_2)$ takes as input the prover's commitments and outputs a value $\alpha \in \mathbb{F}_{q^m}$. The last one $St_2(c_1, c_2, c_3)$ takes as input the prover's commitments and the answer c_3 and outputs the challenge $b \in \{1, 2\}$. The simulator $\mathcal{S}$ will generate a communication transcript representing the interaction between a prover and a verifier. The goal is to generate an indistinguishable communication transcript from a real communication transcript generated by a honest interaction. The simulator $\mathcal{S}$ is defined in three steps:

Step 1. The simulator $\mathcal{S}$ chooses randomly $b \in \{0, 1\}$.

- If $b = 0$, $\mathcal{S}$ chooses randomly $u \in V(k)$, $P \in Gl_{2k}(\mathbb{F}_q)$ and $Q \in Gl_m(\mathbb{F}_q)$ and solves the equation $x = fG \oplus e$ without necessarily satisfying the condition $rank(e) = r$. Then he computes $c_1 = h(P \parallel Q \parallel (uG_1 \oplus f))$ and c_2 is taken as random value. $\mathcal{S}$ simulates the verifier by applying $St_1(c_1, c_2)$ to get $\alpha \in \mathbb{F}_{q^m}$ and then $\mathcal{S}$ can compute $c_3 = h(Q * (uG \oplus (e \bullet \alpha))P)$. Now $\mathcal{S}$ has the needed information to generate a simulated communication data between a prover and a verifier. Therefore the elements to be written in the communication transcript consist of components $C = (c_1, c_2)$, c_3 and $R = (P \parallel Q, (u \oplus f\tilde{\alpha}_k), (uG_1 \oplus f))$. Given the variables used to compute C , c_3 and R were chosen randomly, it follows that the distribution of these elements is indistinguishable from the resulting from a honest interaction.
- If $b = 1$, $\mathcal{S}$ chooses randomly $u \in V(k)$, $P \in Gl_{2k}(\mathbb{F}_q)$ and $Q \in Gl_m(\mathbb{F}_q)$. This time he randomly choses $f \in V(k)$ and $e \in V(2k)$ such that $rank(e) = r$. Then he computes $c_2 = h(Q * (uG)P)$ and c_1 is taken as random value. $\mathcal{S}$ simulates the verifier by applying $St_1(c_1, c_2)$ to get $\alpha \in \mathbb{F}_{q^m}$ and then $\mathcal{S}$ can compute $c_3 = h(Q * (uG \oplus (e \bullet \alpha))P)$. Now $\mathcal{S}$ have the needed information to generate a simulated communication data between a prover and a verifier. Therefore the elements to be written in the communication transcript consist of components: $C = (c_1, c_2)$, c_3 and $R = (Q * (uG)P, Q * (e \bullet \alpha)P)$. Given that the variables used to compute C , c_3 and R were chosen randomly, it follows that the distribution of these elements is indistinguishable from the resulting from an honest interaction.

Step 2. The simulator $\mathcal{S}$ uses the verifiers strategy $St_2(c_1, c_2, c_3)$ to get as result b' .

Step 3. If $b = b'$, the simulator $\mathcal{S}$ writes on its communication transcript the values of C , α , c_3 , b and R . Else, nothing is written and the simulator returns to Step 1.

Therefore, in 2δ rounds on average, $\mathcal{S}$ generates a communication transcript indistinguishable from another communication transcript that corresponds to a honest identification process execution. $\square$

Soundness. Let $\mathbb{F}_{q^m}$ the used finite field. To prove the soundness of our scheme we start to show that for each round a dishonest prover can cheat with a probability that does not exceed $\frac{q^m}{2(q^m-1)}$.

We assume that a dishonest prover can use the following strategies to deal with the challenge that the verifier should send. The first strategy St_0 corresponds to the actions taken by the

dishonest prover when he or she hopes to receive 0 as challenge. He or she randomly chooses u , P and Q and solves the equation $x = f'G \oplus e'$ without necessarily satisfying the condition $\text{rank}(e') = r$ where $f' \in V(k)$ and $e' \in V(2k)$. Then he or she computes $c_1 = h(P \parallel Q \parallel (uG_1 \oplus f))$ and set c_2 at random data. Thus the dishonest prover will be able to answer the challenge $b = 0$ regardless the value of α chosen by the verifier. The second strategy St_1 corresponds to the case where the dishonest prover hopes to receive 1 as challenge. He or she randomly chooses u , P and Q and then randomly generates the couple (f', e') such that $\text{rank}(e') = r$. In this case he or she can compute $c_2 = h(Q * (uG)P)$ and set c_1 at random data. In this case the rank of e' is valid. Thus the dishonest prover can correctly answer the challenge $b = 1$ regardless the value of α .

By trying to guess α , the two strategies can be improved. Let v' the guessed value of α . Thus the dishonest prover can compute $h(v)$ where $v = Q * (uG \oplus e \bullet v')P$.

In St_0 , instead of randomly generating c_2 , the dishonest prover computes $c_2 = h(v \oplus v'')$ where v'' is a random vector of $\text{rank } r$ which will be sent as an answer if $b = 1$ instead of $Q * (e' \bullet v')P$. With this strategy, the dishonest prover can answer to $b = 0$ regardless the value of α chosen by the verifier and to $b = 1$ if $\alpha = v'$.

Let B be a vector in $V(m)$ and n an integer with $m \geq n$, $B_{[n]}$ represents the n last components of B . In St_1 , instead of randomly generating c_1 , the dishonest prover computes

$$c_1 = h(P \parallel Q \parallel (((Q^{-1} * vP^{-1}) \oplus (e' \bullet v'))_{[k]} \oplus f')).$$

With this strategy, the dishonest prover can answer to $b = 1$ regardless the value of α chosen by the verifier and to $b = 0$ if $\alpha = v'$.

Therefore, the success probability of a strategy St for one round is given by:

$$\begin{aligned} P &= \sum_{i=0}^1 P(St = St_i)P(b = i) + P(St = St_i)P(b = 1 - i)P(\alpha = v') \\ &= \frac{q^m}{2(q^m - 1)} \end{aligned} \quad (5.11)$$

We now show that if a dishonest prover successes to cheat with a probability higher than $\left(\frac{q^m}{2(q^m - 1)}\right)^\delta$ where δ is the number of rounds then he or she can solve the Rank Syndrome Decoding Problem (Problem 4).

Theorem 7. *Let ε be a real number > 0 , if $\mathcal{V}$ is an honest verifier who accepts a cheating prover $\mathcal{C}$ proof with probability $\left(\frac{q^m}{2(q^m - 1)}\right)^\delta + \varepsilon$, then there exists a polynomial time probabilis-*

tic algorithm $\mathcal{A}$ which, with a big probability, either computes a valid secret (f, e) or finds a collision for the hash function h . Where h is used to hash the commitments.

Proof. Let Γ be the interaction tree between a cheating prover $\mathcal{C}$ and a verifier $\mathcal{V}$ corresponding to all challenges sended by $\mathcal{V}$ when the adversary has a random transcript τ . The verifier $\mathcal{V}$ has to ask $2(q^m - 1)$ challenges at each round. Each challenge is a couple (α, b) where $\alpha \in \mathbb{F}_{q^m}$ and $b \in \{0, 1\}$. In the following, we will show that a secret key (f, e) can be found from a vertex of $q^m + 1$ children. So we will show that $\mathcal{A}$ can find a such vertex in Γ with a big probability.

Let Λ be a vertex with $q^m + 1$ children. We are in the case where $\mathcal{C}$ is able to answer to $q^m + 1$ queries. It means that for a fixed commitments c_1 and c_2 there exist α such that the cheater answered correctly to the challenges $(\alpha, 0)$, $(\alpha, 1)$. Thus the cheater sends the following answers:

- (v, p, q) for the challenge $(\alpha, 0)$,
- (y, z) for the challenge $(\alpha, 1)$.

where $v = (u \oplus f \tilde{\alpha}_k)$, $p = (uG_1 \oplus f)$, $q = (P \parallel Q)$, $y = Q * (uG)P$ and $z = Q * (e \bullet \alpha)P$ and G_1 is a $k \times k$ matrix over $\mathbb{F}_{q^m}$ such that $G = (I_k, G_1)$. Thus we have either that the cheater can find a collision in hash function h or get the following equations:

$$Q * (vG \oplus (e \bullet \alpha))P = z \oplus y \quad (5.12)$$

$$\text{rank}(z) = r \quad (5.13)$$

from the above equation we get:

$$vG \oplus (Q^{-1} * zP^{-1}) = (Q^{-1} * yP^{-1}) \oplus (x \bullet \alpha) \text{ and } \text{rank}(z) = \text{rank}(Q^{-1} * zP^{-1}) = r.$$

Therefore the values $(Q^{-1} * yP^{-1}) \oplus (x \bullet \alpha)$ and $(v, (Q^{-1} * zP^{-1}))$ can be respectively used as public and private keys to impersonate a legitimate prover.

Now we focus on the probability for Γ to get such vertex of $q^m + 1$ sons. Let τ and Y be respectively a random transcript where $\mathcal{C}$ chooses randomly its values and a non empty set in $\mathbb{F}_{q^m}^* \times \{0, 1\}$. We notice that the sets τ and Y are considered as probability space with uniform distribution. Now let (c, α, b) be the communication data between $\mathcal{C}$ and $\mathcal{V}$ for a complete

identification process where c , α and b are a set of commitments and responses, the set of first challenge and the set of second challenge respectively. We say that (c, α, b) is valid if the identification of $\mathcal{C}$ by $\mathcal{V}$ is a success.

Let us consider Λ a subset of $(\tau \times Y)^\delta$, the set of all possible communication transcript data between $\mathcal{C}$ and $\mathcal{V}$ for a complete identification process, composed by all valid triple (c, α, b) (by $\#E$ we mean the cardinal of the set E). By Theorem 7 assumption we have:

$$\frac{\#\Lambda}{\#((\tau \times Y)^\delta)} \geq \left(\frac{q^m}{2(q^m - 1)} \right)^\delta + \varepsilon \quad (5.14)$$

Let Ω_δ be a subset of τ^δ such that:

- if $c \in \Omega_\delta$, then $(q^m)^\delta + 1 \leq \#\{(\alpha, b) \text{ such that } (c, \alpha, b) \text{ be valid}\} \leq (2(q^m - 1))^\delta$;
- if $c \in \tau^\delta \setminus \Omega_\delta$, then $0 \leq \#\{(\alpha, b) \text{ such that } (c, \alpha, b) \text{ be valid}\} \leq (q^m)^\delta$.

Then $\Lambda = \{ \text{valid } (c, \alpha, b) \text{ such that } c \in \Omega_\delta \} \cup \{ \text{valid } (c, \alpha, b) \text{ such that } c \in \tau^\delta \setminus \Omega_\delta \}$, therefore

$$\#\Lambda \leq \#\Omega_\delta (2(q^m - 1))^\delta + (\#(\tau^\delta) - \#\Omega_\delta) (q^m)^\delta. \quad (5.15)$$

Thus

$$\begin{aligned} \frac{\#\Lambda}{\#((\tau \times Y)^\delta)} &\leq \frac{\#\Omega_\delta}{\#((\tau)^\delta)} + (q^m)^\delta \left((2(q^m - 1))^{-\delta} - \frac{\#\Omega_\delta}{\#((\tau \times Y)^\delta)} \right) \\ &\leq \frac{\#\Omega_\delta}{\#((\tau)^\delta)} + \left(\frac{q^m}{2(q^m - 1)} \right)^\delta. \end{aligned}$$

It follows that

$$\frac{\#\Omega_\delta}{\#((\tau)^\delta)} \geq \varepsilon. \quad (5.16)$$

By (5.16) we have the probability that a cheater prover can answer to at least $(q^m)^\delta + 1$ challenges, by choosing random values, is greater than ε . Whereas, if a cheater prover can answer to more than $(q^m)^\delta + 1$ challenges then we can find at least a vertex with $q^m + 1$ sons at an interaction tree between $\mathcal{C}$ and Λ . Thus it is possible to find an interaction tree with a vertex with $q^m + 1$ sons with a probability close to 1, by resetting and repeating $\lceil \frac{1}{\varepsilon} \rceil$ times the cheater $\mathcal{C}$. Therefore, it is clear that our protocol has a cheating probability too close to $\frac{1}{2}$. $\square$

5.4 Parameters and Results

In this section we present sets of parameters for our identification scheme. Let k, m, q, n and r be integers with $n = 2k$. Our scheme uses as public matrix a random double circulant matrix $G = (I_k | G_1)$ where G_1 is $k \times k$ circulant matrix over $\mathbb{F}_{q^m}$. Then to stock a public matrix G we just need $\log_2(q) \times k \times m$ -bits. For each user, we use as private key the couple (f, e) where $f \in V(k)$ and $e \in V(2k)$ such that $\text{rank}(e) = r$ and as public key we use $x \in V(2k)$ such that $x = fG \oplus e$. Thus to stock a public key we need $2 \times k \times \log_2(q)$ -bits and to stock private key we need $3 \times k \times \log_2(q)$ -bits. All sets of parameters proposed lie on the Gilbert-Varshamov bound for rank metric codes [99]. Also the proposed sets of parameters deal with the complexity of the best known attack on Rank Syndrome Decoding problem [7] and the complexity of the attack proposed in [67], because in the case of double circulant code we have always $\lceil \frac{(k+1)(r+1)-(n+1)}{r} \rceil \leq k$, which is about

$$\min(\mathcal{O}\left((n-k)^3 m^3 q^{r \lceil \frac{(k+1)m}{n} \rceil - m}\right), \mathcal{O}\left(k^3 r^3 q^{r \lceil \frac{(k+1)(r+1)-(n+1)}{r} \rceil}\right)).$$

The first one is to get a security of 2^{80} for a standard using. The second set of parameters is to get a security of 2^{128} . We propose these latest parameters to prevent the fast evolution of computer processors.

Now let l_h be the length of the hash function h and δ be the number of rounds of our protocol. To authenticate the prover in δ rounds, the prover and the verifier have to exchange:

$$\delta \times \left(1 + l_h + \log_2(q) \times m + \frac{(6 \times m \times k + 2 \times k + m) \times \log_2(q)}{2}\right) + 2 \times l_h\text{-bits}$$

To have a cheating probability of 2^{-16} with a security of 80 bits we propose the following parameters:

$$q = 2, m = 23, k = 12, r = 7, \delta = 16 \text{ and } l_h = 160. \quad (5.17)$$

In this case the known attacks on Rank Syndrome Decoding problem lead to a complexity of at least 2^{92} operations and we get:

- The needed space to store the public matrix is 276 bits,
- the needed space to store the public or private key is 552 bits,
- and the total data exchanged to authenticate a prover is 16888 bits.

	Rounds	Matrix(bit)	Public Key(bit)	Secret Key(bit)	communication(bit)
Stern [123]	28	122500	350	750	42019
Veron [127]	28	122500	700	1050	35486
CVE [40]	16	32768	512	1024	31888
AGS [1]	18	350	700	700	20080
Stern Rank [69]	28	1980	180	400	22400
RankId	16	276	552	552	16888

Table 5.1: Comparison between different code-based Zero-knowledge schemes for a 2^{-16} cheating probability with a security of 2^{80}

	Rounds	Matrix(bit)	Public Key(bit)	Secret Key(bit)	communication(bit)
CVE [40]	16	86528	832	1664	47248
AGS [1]	18	550	1100	1100	32304
RankId	16	434	868	868	26064

Table 5.2: Comparison between different code-based Zero-knowledge schemes for a 2^{-16} cheating probability with a security of 2^{128}

We also propose the following parameters to get a cheating probability of 2^{-16} with a security of 128 bits:

$$q = 2, m = 31, k = 14, r = 8, \delta = 16 \text{ and } l_h = 236. \quad (5.18)$$

In this case the known attacks on Rank Syndrome Decoding problem lead to a complexity of at least 2^{131} operations and we get:

- The needed space to store the public matrix is 434 bits,
- the needed space to store the public or the private key is 868 bits,
- and the total data exchanged to authenticate a prover is 26064 bits.

To end this section we show in Tables 5.1 and 5.2 the performances of our scheme compared to existing code-based Zero-knowledge identification protocols.

Through the practical results presented in Tables 5.1 and 5.2, it is easy to see that our Zero-knowledge identification protocol is the most efficient from code-based ones. Indeed, in the case of the parameters for a cheat probability of 2^{-16} with a security of 2^{80} (Table 5.1), the storage of the public matrix, the public key and the secret key requires 1380 bits in our protocol and 1750 bits in the best Zero-knowledge code-based authentication protocol [1], which represents a gain of 21%. Then, the size of the communication data needed to authenticate

a prover is 16888 bits in our protocol and 20080 bits in the best code-based Zero-knowledge identification protocol [1], which represents a gain of 16%. In addition, in the case of parameters for a cheat probability of 2^{-16} with a security of 2^{128} (Table 5.2), the storage of the public matrix, the public key and the secret key requires 2170 bits in our protocol and 2750 bits in the best Zero-knowledge code-based authentication protocol [1], which represents a gain of 21%. Next, the size of the communication data needed to authenticate a prover is 26064 bits in our protocol and 32304 bits in the best code-based Zero-knowledge protocol [1], which represents a 19% gain.

Computational cost. We divided the computational cost in two parts. the first part is the verifier's computation which is about $2^{11.5}$ operations in $\mathbb{F}_q$. The second part is the prover's computation which is about $2^{14.9}$ operations in $\mathbb{F}_q$ while in AGS identification scheme [1] the prover's computation is about 2^{21} in $\mathbb{F}_2$.

Signature. Let n be an integer ≥ 1 . In [4] Dagdelen *et al.* gave a generic methodology to prove the security of signature scheme obtained by applying the Fiat-Shamir paradigm [62] to $(2n + 1)$ -pass identification scheme. Then we can apply in safe the Fiat-Shamir paradigm to our five-pass Zero-knowledge identification scheme to obtain a signature scheme. In our case to generate a signature for a security of 2^{80} , we use parameters presented by expression (5.17) and we perform 80 rounds to keep a cheating probability of 2^{-80} . Thus we get a signature size of 83160 bits which is better than AGS [1] case where the signature size is about 93000 bits.

5.5 Conclusion

In this chapter, we have proposed an efficient Zero-knowledge identification scheme based on coding theory assumptions which is supposed to be resistant to quantum computers. We have used rank metric rather than the classical Hamming metric. This choice implies a significant reduction in public data size and in communication cost. For instance, we achieve a gain of 47% in our protocol if compared to AGS identification scheme [1] for a 80 bits security. Our scheme presents optimal results in comparison with all code based identification schemes. It is also possible to turn it into a signature scheme by applying the Fiat-Schamir Transformation described in [62].

Rank-Metric Code-Based Mutual Authentication Protocol for RFID

Contents

6.1	Introduction	84
6.2	The Proposed Scheme	87
6.2.1	Description of the Proposed Scheme	88
6.2.2	Security Analysis	89
6.3	Practical Results and Comparison	93
6.4	Conclusion	96

In this chapter based on [16], we present the first mutual authentication protocol for RFID systems based on hard problems from rank metric codes, and we discuss its security in the post quantum scenario. Furthermore, our protocol uses two improvements. The first one consists in using double circulant low rank parity check codes and permits to reduce communication costs and to get smaller key size than previous code-based protocols. The second one consists in using a hash function to get the full inside secrecy property. Moreover, our protocol saves at least 42% in terms of communication data compared to the most recent code-based authentication protocols for RFID systems.

6.1 Introduction

Thanks to their small size, low cost, low voltage and long lifetime, Radio Frequency Identification (RFID) systems are becoming increasingly popular and are used more and more frequently in applications such as access control, supply chain management, e-health and so on. RFID is an automatic identification technology that uses radio frequency radiation to identify tagged objects. We consider RFID system with Reader and Tag only, described as follows:

Tag, denoted as T , it is a small electronic chip supplemented with an antenna that can transmit and receive data. RFID Tag can be divided into three categories: passive Tag, which is powered by the radio wave from an RFID Reader and communicates with the Reader

through backscattering. Active and semi-passive RFID Tags use internal batteries to power their circuits. An active Tag also uses its battery to broadcast radio waves to the Reader, whereas a semi-passive Tag relies on the Reader to supply its power for broadcasting.

Reader, denoted as R , it is a device to communicate with Tag by radio waves.

The authentication process of RFID Tags by an RFID Reader is necessary to ensure the authenticity, the freshness and the presence of the RFID Tags. During this authentication process, RFID Tags can check the authenticity, the freshness and the presence of the Reader, which is so-called mutual authentication.

The main issues in RFID systems are the constraint of resources and the security requirements. In addition to anonymity and untraceability, an authentication protocol for RFID systems must verify the following six properties as cited in [47] namely: Mutual Authentication, Desynchronization Resilience, Anonymity, Untraceability, Forward Secrecy and Resistance to Replay Attack. Our protocol also verifies the Full Inside Secrecy property, which is an extension of the Inside Secrecy property presented in [10]. We define the Full Inside Secrecy property as follow:

Full Inside Secrecy means that even if we have a genuine corrupted Reader with some secrets, an adversary cannot extract the private keys from the Tags. Furthermore if an adversary can corrupt one or many Tags, he or she cannot produce a Reader which can be authenticated by a Tag which has not been corrupted.

To figure out the importance of this property, consider a practical example of a shop where each article is equipped with a Tag. In this case no one should be able to pick up an article and get out from the shop without paying it. But in various authentication schemes, if someone can get access to a Tag memory then he or she can get the necessary information to produce a reader which can be accepted by another Tag and then get the article marked as paid. The Full Inside Secrecy property gives a direct response to the previous issue by ensuring that even if an adversary can corrupt one or many Tags, he or she cannot produce a Reader which can be authenticated by a Tag which has not been corrupted. The Full Inside Secrecy property also ensures the Inside Secrecy property, which means that even if we have a vulnerable Reader, an adversary cannot obtain any more Tag's identifier since this identifier has been never revealed by the Tag in the authentication process.

By assuming the existence of a quantum computer, Shor's algorithm (1994) showed that the discrete logarithm and the factorization problems can be solved in polynomial time. Which enabled the emergence of post-quantum cryptography and especially code-based cryptography for RFID identification system.

In the literature there exist four types of RFID authentication protocols. The first one consists of protocols that support cryptographic functions whether symmetric or asymmetric [101]. The second type is composed of protocols that use pseudo random generator number PRNG and hash functions [10, 52, 82]. The third type of protocols called lightweight protocols is generally based on some error correcting codes such as Cyclic Redundancy Check (CRC) and checksum [17, 106, 129]. The last type of protocols called ultra-lightweight are those protocols that use only basic binary operations like XOR, AND and NOT [81, 122].

Over the years, many authentication protocols for RFID systems have been proposed and the majority of them are based on number theory and elliptic-curve cryptography [43, 106]. Hung-Yu [82] presented in 2006 an RFID authentication protocol which provides the anonymity property besides other properties. Later, Naved *et al.* [106] presented a reliable and low cost RFID authentication protocol. In [50] Cui *et al.* proposed a code-based authentication which does not achieve forward secrecy property. Thereafter, in [46] Chien and Laih used a confusion scheme to ensure the untraceability of their protocol based on error correcting codes. However, it does not achieve the inside secrecy property. In [100] Malek *et al.* based their authentication protocol on randomized McEliece cryptosystem, however this scheme cannot achieve desynchronization resilience. In [92], a mutual authentication code-based protocol is presented by Li *et al.* However, this scheme is not resistant to traceability and forward secrecy attacks. A year later, this scheme was improved by Chikouche *et al.* [47] to achieve untraceability and forward secrecy properties. In [10], two mutual authentication protocols using code-based Zero-Knowledge identification scheme proposed by Assidi *et al.* [10], that achieved Inside Secrecy. However an inconvenience of [10] is the fact that it is possible to produce a reader which can be authenticated by a tag which is not corrupted. With the hypothetical deployment of quantum computers, code-based cryptography is a valid alternative to number theoretical cryptography. Besides, code-based cryptography is used in many cryptographic primitives such as identity-based cryptography, group signature and ring signature [9, 14, 66, 104].

In this chapter, we present the first mutual authentication protocol for RFID systems based on rank metric codes which does not use Zero-Knowledge identification schemes as in [10]. Furthermore, our protocol uses two improvements, the first one consists in using double circulant low rank parity check codes. This improvement permits to reduce communication costs and

to get smaller key size than previous code-based protocols. The second security enhancement consists in using a hash function to get full inside secrecy property and the following properties: mutual authentication, desynchronization resilience, anonymity, untraceability, forward secrecy and resistance to replay attacks.

The remainder of this chapter is organized as follows: we describe our scheme in Section 6.2 and analyze its security, Section 6.3 is devoted to practical results and comparison of our protocol with other known schemes. Finally, we conclude our chapter in Section 6.4.

6.2 The Proposed Scheme

In this section, we first describe our RFID authentication protocol (Algorithm 6.1). This protocol achieves the mutual authentication property, using a rank metric code, elementary operations like bitwise XOR, bit shift and a hash function to hide the tag identity, which ensures the Full Inside Secrecy property. Then we analyze the security of our scheme in the second part of this section.

Let q be a power of a prime p and m an integer > 0 . We denote by $GL_m(\mathbb{F}_q)$ the group of invertible $m \times m$ matrices over $\mathbb{F}_q$. Let $\beta = (\beta_1, \dots, \beta_m)$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Let $\mathcal{F}_i$ be the map from $\mathbb{F}_{q^m}$ to $\mathbb{F}_q$ where $\mathcal{F}_i(x)$ is the i^{th} coordinate of x in the basis β . For any $v = (v_1, \dots, v_n)$ in $\mathbb{F}_{q^m}^n$ we construct the matrix $\bar{v} \in \mathcal{M}_{m,n}(\mathbb{F}_q)$ where

$$\bar{v} = (\bar{v}_{i,j}) = \mathcal{F}_i(v_j) \quad (6.1)$$

The rank weight of a vector v denoted $rank(v)$ is defined as the rank of the associated matrix $\bar{v}$ defined by (6.1). For a given basis β , we denote by Φ_β the inverse computed in the basis β of the map

$$\mathbb{F}_{q^m}^n \rightarrow \mathcal{M}_{m,n}(\mathbb{F}_q) : v \mapsto \bar{v} \quad (6.2)$$

.

We following recall a decisional problem for rank metric codes.

Problem 5 (Decisional Rank Syndrome Decoding Problem(DRSD)). *Let G be a full-rank $k \times n$ matrix over $\mathbb{F}_{q^m}$, $v \in \mathbb{F}_{q^m}^k$ and $x \in \mathbb{F}_{q^m}^n$ of weight r . Can we distinguish the pair $(G, v \cdot G + x)$ from (G, y) with random vector $y \in \mathbb{F}_{q^m}^n$?*

In [66] Gaborit *et al.* proved that the DRSD problem is hard in the worst case. In our proposed protocol, Problem 3 is used to prove the anonymity property and Problem 5 is used to prove the untraceability property, while Problem 4 is used to prove the mutual authentication and the full inside secrecy properties.

6.2.1 Description of the Proposed Scheme

Our RFID system is composed of many Readers and Tags such that:

- Each Tag and Reader has a Pseudo Random Number Generator (PRNG) function.
- Each Tag and Reader can generate random vector of $\mathbb{F}_{q^m}$ of a fixed length with arbitrary rank weight.

Let m, q, n, k, r and d be integers with $n = 2k$. We now begin the description of our RFID authentication protocol with the initialization phase to finish with the authentication phase.

Initialization. In the initialization phase, we first choose randomly a DC-LRPC code $\mathcal{C}$ over $\mathbb{F}_{q^m}$ of length n , dimension k and minimal rank distance d which can correct error of rank r . The code $\mathcal{C}$ has a parity check matrix $H \in \mathcal{M}_{n \times k}(\mathbb{F}_{q^m})$ and a generator matrix $G \in \mathcal{M}_{k \times n}(\mathbb{F}_{q^m})$. We compute G in systematic form which means that $G = (G_1 | I_k)$ where G_1 is a $k \times k$ circulant matrix over $\mathbb{F}_{q^m}$. Now, let g_1 be the first row vector of G_1 and let $U \subset \mathbb{N}$ be a set of Tag's index. Thus, for each $i \in U$, to initialize the Tag T_i , we randomly choose a unique vector $v_i \in \mathbb{F}_{q^m}^k$ and store v_i, g_1 and h (a hash function) in each Tag T_i . At the end of this phase, we initialize the Reader by creating a dataset

$$\Omega = \{h(v_i)\}_{i \in U} \quad (6.3)$$

and storing Ω, H and G in the Reader.

Authentication. Let us consider a Reader R , a Tag T and let B be a vector in $\mathbb{F}_{q^m}^n$ and k_1, k_2 integers such that $0 \leq k_1 < k_2 \leq n$. Let $B_{[k_1:k_2]}$ denotes all the components of B that are between the indexes k_1 and k_2 where k_1 is included but k_2 is not. During this step, the Reader authenticates the Tag first and then the Tag's authenticate the Reader. We perform the authentication phase using the steps described in Algorithm 6.1 where notations are as above.

At the last step of Algorithm 6.1, it is easy to verify that if

$$C_2 := u \cdot H \oplus ((\overline{N_R * N_T}) * N_R)$$

then $G \cdot C_2 = G \cdot ((\overline{N_R * N_T}) * N_R)$ where $H = (H_1 | H_2)$, $G = (H_1^{-1} \cdot H_2 | I_k)$ and $H_1, H_2 \in \mathcal{M}_{k \times k}(\mathbb{F}_{q^m})$. Indeed, we have :

$$\begin{aligned} G \cdot C_2 &= G \cdot (u \cdot H \oplus ((\overline{N_R * N_T}) * N_R)) \\ &= (G \cdot u \cdot H) \oplus G \cdot ((\overline{N_R * N_T}) * N_R) \end{aligned}$$

and

$$\begin{aligned} G \cdot u \cdot H &= (H_1^{-1} \cdot H_2 | I_k) \cdot u \cdot (H_1 | H_2) \\ &= (H_1^{-1} \cdot H_2 | I_k) \cdot (u \cdot H_1 | u \cdot H_2) \\ &= (u \cdot H_1) \cdot (H_1^{-1} \cdot H_2) \oplus (u \cdot H_2) \cdot I_k \\ &= 0 \end{aligned}$$

then it is clear that

$$G \cdot C_2 = G \cdot ((\overline{N_R * N_T}) * N_R) \quad (6.4)$$

6.2.2 Security Analysis

In this section, we focus on the security of the proposed scheme (Algorithm 6.1). The security analysis consists in proving the following seven security properties: anonymity, untraceability, mutual authentication, resistance to the replay attack, resistance to the desynchronizing attack, forward secrecy and full inside secrecy.

Anonymity: Our protocol uses a unique identifier v for a Tag T . Thus, to break the anonymity of our scheme, an adversary $\mathcal{A}$ has to either solve the RSD problem (Problem 3) or to guess $v \in \mathbb{F}_{q^m}^k$. However, this event may occur with at most negligible probability $\frac{1}{q^{k \times m}}$.

Untraceability: Let $\mathcal{A}$ be an adversary against the untraceability property presented in [110] with an advantage ε . In the following, we prove that ε is negligible using the hardness of the DRSD problem (Problem 5). We consider the experiment $\mathcal{G}_0^b$ which depends on the bit b and the experiment $\mathcal{G}_1$:

Experiment $\mathcal{G}_0^b$. It is the original untraceability experiment presented in [110]. After the learning and the challenge phases, since the decoding algorithm can fail, with a probabil-

Algorithm 6.1 Our RFID Authentication Scheme

1. The Reader generates a random vector N_R of $\mathbb{F}_{q^m}^n$ such that $\text{rank}(N_R) < r$ and sends N_R to a Tag T_i .
2. The Tag generates a random vector N_T of $\mathbb{F}_{q^m}^n$ of length n such that $\text{rank}(N_T) < r$ and computes:

$$C_1 := (h(v_i) \oplus (\overline{N_R} * N_T)_{[0:k]}) \cdot G \oplus (\overline{N_R} * N_T)$$

and sends C_1 to the Reader. (The bar symbol $\bar{\cdot}$ is as in (6.1) and the $*$ -product is as in Definition 8.)

3. The Reader uses his secret key H through the efficient decoding algorithm to retrieve $h(v_i)$ and $\overline{N_R} * N_T$ from C_1 . If the decoding algorithm fails (LRPC Codes can have decoding failure) then the Reader set C_2 to a random vector of $\mathbb{F}_{q^m}^n$. Else we have two cases:
 - if $h(v_i)$ is in the dataset Ω then the Tag is authenticated and the Reader generates a random vector $u \in \mathbb{F}_{q^m}^k$ and computes

$$C_2 := u \cdot H \oplus ((\overline{N_R * N_T}) * N_R)$$

• else the Tag is not authenticated and the Reader sets C_2 to a random vector of $\mathbb{F}_{q^m}^n$. Then it sends C_2 to the Tag.

4. To authenticate the Reader, the Tag checks if

$$G \cdot ((\overline{N_R * N_T}) * N_R)$$

is equal to the received $G \cdot C_2$ (see (6.4)).

ity less than $q^{(1-r)(d-2)}$, then with a probability larger than $1 - q^{(1-r)(d-2)}$ the adversary A gets: N_R ,

$$C_1 = (h(v_b) \oplus (\overline{N_R} * N_T)_{[0:k]}) \cdot G \oplus (\overline{N_R} * N_T) \quad (6.5)$$

and

$$C_2 = u \cdot H \oplus ((\overline{N_R * N_T}) * N_R). \quad (6.6)$$

In the Guess phase, $\mathcal{A}$ outputs b with a probability $> \frac{1}{2}$. Where $v_b \in \mathbb{F}_{q^m}^k$ is the *ID* of the Tag T_b .

Experiment $\mathcal{G}_1$. Let R be a random vector of length n over $\mathbb{F}_{q^m}$. In this experiment, we just modify the generation of C_1 as follow: $C_1 = R$.

Now we show that if $\mathcal{A}$ can distinguish Experiment $\mathcal{G}_0^b$ from $\mathcal{G}_1$, then there exists a distinguisher $\mathcal{D}$ solving the DRSD problem (Problem 5). Given a vector C^* , the goal of $\mathcal{D}$ is to guess whether C^* is random or C^* is generated by the encryption of a vector using a McEliece-like cryptosystem described in Subsection 2.2. We give to $\mathcal{D}$ challenges C^* and the ability to interact with $\mathcal{A}$. The distinguisher $\mathcal{D}$ interacts with $\mathcal{A}$ as follow:

Setup Execute the setup and send Ω and G to $\mathcal{A}$.

Challenge 1. Set $C_1^* = C^*$

2. Output (N_R, C_1^*, C_2) .

It is clear that if C_1^* is generated by the encryption of a vector using a McEliece-like cryptosystem described in Subsection 2.2, then the view of $\mathcal{A}$ after its interaction with $\mathcal{D}$ is close to $\mathcal{G}_0^b$. On the other hand, if C_1^* is random vector then the view of $\mathcal{A}$ is close to $\mathcal{G}_1$. We stress that in $\mathcal{G}_1$ the answer to the challenge bit b is computed correctly as in experiment $\mathcal{G}_0^b$. thus the advantage of $\mathcal{A}$ in $\mathcal{G}_1$ is null. That implies the advantage of $\mathcal{A}$ in attacking the untraceability property is negligible.

Mutual authentication: We consider five integers n_1, n_2, n_3, n_4 and n_5 polynomially bounded in the security parameter. To study Mutual authentication property, we use the following phases:

Learning: In this phase, $\mathcal{A}$ observes the protocol running n_1 times, interacts n_2 times with the Reader and interacts n_3 times with the Tag.

Challenge: In this phase, $\mathcal{A}$ tries to impersonate n_4 times the Reader and n_5 times the Tag.

Let $\mathcal{E}_1$ be the event where $\mathcal{A}$ successfully impersonates the Tag which means that $\mathcal{A}$ guesses the good Tag's ID ($v \in \mathbb{F}_{q^m}^k$) in one of the n_5 sessions during the challenge phase or solves the RGD problem (Problem 4). Thus, $\mathcal{E}_1$ may occur with at most negligible probability $\frac{n_5}{q^{mk}}$. Now let $\mathcal{E}_2$ be the event that $\mathcal{A}$ successfully impersonates the Reader, which means that $\mathcal{A}$ guesses the Reader's secret key ($H \in \mathcal{M}_{k \times n}(\mathbb{F}_{q^m})$) in one of the n_4 sessions during the challenge phase or solves the RGD problem (Problem 4). Thus, $\mathcal{E}_2$ may occur with at most negligible probability $\frac{n_4}{q^{mn}}$.

Desynchronizing attack: We assume that an adversary $\mathcal{A}$ tries to desynchronize our protocol execution by modifying or cutting the communication data between the Tag and the Reader in a current session. The authentication in this session will not succeed. However, in the next session a normal execution of our protocol will succeed because we do not update any internal state.

Replay attack: By replaying the same message from previous communication between the Reader and the Tag, the adversary $\mathcal{A}$ has to impersonate the Reader or the Tag. At the first time we assume that $\mathcal{A}$ tries to impersonates the Tag. After receiving N_R , he or she picks a vector C_1 from previous legitimate communication. We say that $\mathcal{A}$ has successfully impersonates the Tag if $\mathcal{A}$ has the knowledge of v or the Reader uses a repeated $N_R \in \mathbb{F}_{q^m}^n$. This event may occur with at most negligible probability $\frac{1}{q^{mn}}$. At the last time we assume that $\mathcal{A}$ tries to impersonate the Reader. After receiving C_1 , he or she picks a vector C_2 from previous legitimate communication. We say that $\mathcal{A}$ has successfully impersonate the Reader if $\mathcal{A}$ has the knowledge of H or the Tag uses a repeated $N_T \in \mathbb{F}_{q^m}^n$ such that $\text{rank}(N_T) < r$. This event may occur with at most negligible probability.

Forward secrecy: To prove the forward secrecy of the proposed protocol, we assume that an adversary $\mathcal{A}$ who can get access to the Tag's memory, and we show that $\mathcal{A}$ cannot compromise the previous communication. Let (N'_R, C'_1, C'_2) be the communication transcript between the Tag and the Reader. Indeed, because C'_1 and C'_2 are computed using N'_R and N'_T which are picked randomly, it is clear that even if $\mathcal{A}$ has access to the Tag's secret key he or she cannot compromise the previous communication.

Full Inside Secrecy: Let n_1, n_2, n_3, n_4, n_5 and n_6 be six integers polynomially bounded in the security parameter. We prove this property using the two following distinct cases:

1. Assuming that an adversary $\mathcal{A}$ has access to the Reader memory i.e. an adversary $\mathcal{A}$ gets Ω as in (6.3) and $H \in \mathcal{M}_{k \times n}(\mathbb{F}_{q^m})$ where H is a parity check matrix. By getting H and ω , the Adversary ($\mathcal{A}$) can just retrieve $h(v)$. We study this first case using the following phase:

Learning: The adversary $\mathcal{A}$ observes n_1 times the protocol running and interacts n_2 times with the Reader R.

Challenge: By n_3 times, $\mathcal{A}$ tries to impersonate the Tag T.

We say that $\mathcal{A}$ has successfully impersonated the Tag T if $\mathcal{A}$ recovers the Tag's ID ($v \in \mathbb{F}_{q^m}^k$) from $h(v)$ or $\mathcal{A}$ guesses Tag's ID ($v \in \mathbb{F}_{q^m}^k$) in one of the n_3 sessions during the challenge phase. This last event may occur with at most negligible probability $\frac{n_3}{q^{mk}}$.

2. Let T' be a corrupted Tag. Assuming that an adversary $\mathcal{A}$ has access to the memory of T' . We study this last case using the following phase:

Learning: The adversary $\mathcal{A}$ observes n_4 times the protocol running and interacts n_5 times with the Tag T' .

Challenge: By n_6 times, $\mathcal{A}$ tries to impersonate the Reader R .

We say that $\mathcal{A}$ has successfully impersonated the Reader R if $\mathcal{A}$ guesses the Reader's secret key ($H \in \mathcal{M}_{k \times n}(\mathbb{F}_{q^m})$) in one of the n_6 sessions during the challenge phase or solves the RGD problem (Problem 4). However, this event may occur with at most negligible probability $\frac{n_6}{q^{mn}}$.

6.3 Practical Results and Comparison

In this section, we evaluate the performance of the proposed RFID mutual authentication protocol in terms of security, memory storage needed and communication data. In Table 6.1, we present a comparison between the proposed RFID mutual authentication protocol and some other known RFID code-based authentication protocols. Table 6.1 shows that our proposed scheme fulfills all the security requirements, namely anonymity, untraceability, mutual authentication, resilience to resynchronization attacks, replay attack, secrecy, forward secrecy and full inside secrecy property.

	[100]	[92]	[47]	[98]	[10]	Our scheme
Anonymity	Y	N	Y	Y	N	Y
Untraceability	Y	N	Y	Y	N	Y
Mutual authentication	Y	Y	Y	Y	Y	Y
Desynchronizing attack	N	Y	Y	Y	Y	Y
Replay attack	Y	Y	Y	Y	Y	Y
Forward secrecy	Y	N	Y	Y	Y	Y
Full Inside secrecy	N	N	N	N	N	Y

Table 6.1: Comparison of security requirements

Remark 5. For a code $\mathcal{C}$ over $\mathbb{F}_{q^m}$ of length n , dimension k and r the rank weight of the error to be decoded, we consider the similar approach in [23] to compute complexity of RSD, i.e.,

$$\mathcal{O}((n-k)^3 m^3 q^{r(k+1)m/n-m}) \quad (6.7)$$

which, is the best known attack, and was obtained [23] by applying Grover's algorithm by square rooting the exponential term in the complexity, see [27]. This, using a classical computer; and the complexity is of

$$\mathcal{O}\left((n-k)^3 m^3 q^{r\frac{(k+1)m}{2n}-m}\right) \quad (6.8)$$

using a quantum computer by applying Grover's algorithm again. Thus, to obtain a good level of security, the quantities q^{mk} and q^{mn} must be large.

An algebraic attack on rank metric code-based cryptosystems is considered recently in [18]. And more recently in [19], by an algebraic approach, Bardet et al. showed that, for a code $\mathcal{C}$ over $\mathbb{F}_{q^m}$ of length n , dimension k , r the rank weight of the error to be decoded and ω is the constant of linear algebra, attacks on RSD problem are in complexity

$$\mathcal{O}\left(m \binom{n-k-1}{r} \binom{n}{r}^{\omega-1}\right) \quad \text{if } m \binom{n-k-1}{r} \geq \binom{n}{r} - 1, \quad (6.9)$$

when taking the maximal value of p such that $m \binom{n-p-k-1}{r} \geq \binom{n-p}{r} - 1$ the complexity is reduced to

$$\mathcal{O}\left(m \binom{n-p-k-1}{r} \binom{n-p}{r}^{\omega-1}\right) \quad (6.10)$$

operations in the field $\mathbb{F}_q$. And

$$\mathcal{O}\left(q^{ar} m \binom{n-k-1}{r} \binom{n-a}{r}^{\omega-1}\right) \quad \text{if } m \binom{n-k-1}{r} < \binom{n}{r} - 1 \quad (6.11)$$

where the parameter a is the number of columns of the matrix (of coordinates of the error e in the basis of the support of e in $(\mathbb{F}_q^m)^r$) where the exhaustive search is performed. The parameter p is used when the system is highly overdetermined, and we do not need all the equations to get the solution, we can use punctured codes to get less equations and less variables, but still stay in the overdetermined case and p corresponds to the number of columns on which the code is punctured.

Usually by [19], in the case of $q = 2$, let b be the smallest positive integer such that $0 < b < r + 2$ and satisfying $A_b - 1 \leq B_b + C_b$ where $A_b = \sum_{j=1}^b \binom{n}{r} \binom{mk+1}{j}$, $B_b = \sum_{j=1}^b \left(m \binom{n-k-1}{r} \binom{mk+1}{j} \right)$ and $C_b = \sum_{j=1}^b \sum_{i=1}^j \left((-1)^{i+1} \binom{n}{r+i} \binom{m+i-1}{i} \binom{mk+1}{j-i} \right)$, then the complexity becomes

$$\mathcal{O}((B_b + C_b)A_b^{\omega-1}) \quad (6.12)$$

We discuss the performance of our schemes in two parts. The first one is about the memory storage needed by the Tag and the second part treats the amount of communication data exchanged during the authentication process between a Tag and a Reader.

Remark 6. When circulant matrices are chosen, the system might be subjected to attack based on projecting and folding [79]. To counter this attack, the choice of k should be chosen such that it is a prime and $x^k - 1$ can be factored into 2 irreducible factors over $\mathbb{F}_k$. Therefore, the choice of $k = 41$ (for 80 security level) is subject to this attack. Nevertheless, the choice of, e.g., $k = 53$ is suitable.

Memory storage: In the proposed scheme, for the Tag T we need to store the Tag's identity v which is of size k bits, the first line g_1 of the public matrix which is of size k bits and the hash function h .

Finally, we generate two sets of parameters for DC-LRPC codes based on attacks' complexity presented in Remark 5. The first one is to get a security of 2^{80} for a standard use. The second set of parameters is to get a security of 2^{128} . We propose these latest parameters to prevent the fast evolution of computer processors. To have a failure probability of 2^{-36} with a security of 80 bits using a classical computer and a security of 53 bits using a quantum computer we propose the following parameters:

$$q = 2, m = 23, k = 61, d = 8 \text{ and } r = 7. \quad (6.13)$$

In this case, the memory storage needed is 2806 bits. We also propose the following parameters to get a failure probability 2^{-25} with a security of 128 bits using a classical computer and a security of 58 bits using a quantum computer:

$$q = 2, m = 47, k = 53, d = 7 \text{ and } r = 6. \quad (6.14)$$

In this case, the memory storage needed is 4982 bits.

Communication cost: The communication cost is the amount of data exchanged between a Tag and a Reader during one execution of the protocol. In our proposed protocol, the Tag just needs to send a vector C_1 of size n bits and the Reader needs to send two vectors N_R of size n and C_2 of size n . Then the total communication cost is about $3n$ bits.

In Table 6.2, we are comparing our scheme for 80 bits of security to the last code-based RFID protocols in [10] and with the RFID protocols based on QC-MDPC codes [98] which can be seen as Hamming version of DC-LRPC codes. Table 6.2 shows that our scheme needs less memory storage than the protocol proposed in [98] which is important because of the limited capacity of RFID devices. Furthermore, our protocol saves respectively 61% and 42% in terms of communication data compared to the protocol proposed in [10] and [98].

	Tag's storage	communication cost
Zhuohua <i>et al.</i> [98]	4928	14656
Assidi <i>et al.</i> [10]	1386	21920
The Proposed scheme	2806	8418

Table 6.2: Comparison of storage and communication performance (in bit)

6.4 Conclusion

In this chapter, we have proposed a mutual authentication protocol based on hard problems from double circulant low rank parity check codes. These problems are supposed to be resistant to quantum computing attacks. Our proposed protocol meets all important security and privacy requirements. In addition to classic security and privacy requirements for RFID systems, our scheme achieves the full inside secrecy property. Also, our protocol saves at least 42% in terms of communication data compared to the most recent code-based authentication protocols for RFID systems.

A Code Based Cryptographic Accumulator and Fully Dynamic Group Signature

Contents

7.1	Introduction	98
7.2	Preliminaries	100
7.2.1	Code-based Cryptography	100
7.2.2	Zero-Knowledge Argument Systems and Stern-Like Protocols . . .	101
7.2.3	Collision-Resistant Hash Function	102
7.3	The Proposed Code Based Cryptographic Accumulator	103
7.3.1	A Code Based Difficult Problems	104
7.3.2	A Family of Code-Based Collision-Resistant Hash Functions	105
7.3.3	The Proposed Code-based Accumulator	107
7.3.4	Example	109
7.4	Security and Implementation of the Proposed Cryptographic Accumulator .	111
7.4.1	Security of the Proposed Cryptographic Accumulator	111
7.4.2	Implementation	115
7.5	The Proposed Fully Dynamic Group Signature scheme	116
7.5.1	Logarithmic-Size Arguments of Set Membership	116
7.5.2	An Update Algorithm	119
7.5.3	Description of Our <i>FDGS</i> Scheme	120
7.5.4	Analysis of Our <i>FDGS</i> Scheme	126
7.5.5	Efficiency	131
7.6	Conclusions	132

we propose a code-based cryptographic accumulator that is quantum computer resistant. Specifically, our scheme is based on the hardness of the Syndrome Decoding problem and satisfies

the collision freeness and indistinguishability requirements. We also use a double circulant codes which allows us to get a small key size. Furthermore, we use proposed cryptographic accumulator to create a fully dynamic code based group signature. Moreover, we give an implementation of our scheme which is, as far as we know, the first direct implementation of a post-quantum cryptographic accumulator.

7.1 Introduction

The cryptographic accumulator notion was first introduced by Benaloh and De Mare [24] in 1993. A cryptographic accumulator is a cryptographic primitive which allows accumulating a finite set of elements into a single value that is called *Accumulated Value*. Therefore, for each element, it is possible to generate a short membership proof called Witness, which attests the belonging of the element to the set. The accumulated value is public. For a cryptographic accumulator to be secure, it should not be possible to find a valid witness for an element not belonging to the set. In the classical example, to accumulate a finite set $X = \{x_i\}$ of big prime integers, we compute the product x of all elements of X . To prove the membership of an element x_i of X , we just compute $q_i = \frac{x}{x_i}$ where q_i is an integer and is called the witness of x_i . Then everyone can check the membership of x_i by checking if $x_i q_i = x$ which is an easy computation. But the factorization of x to extract x_i is hard for usual computers but not for the hypothetical quantum computer.

In the first scheme introduced by Benaloh *et al.* [24], it is not possible to add or remove elements from the set of accumulated values. This type of accumulator is said to be static. Thereafter to overcome this lack, in 2002 Camenisch and Lysyanskaya [35] proposed a first dynamic accumulator based on RSA assumptions which allow elements to be added or removed from the set of accumulated values, but they do not have an efficient non-membership proof. Later, many other constructions have been proposed like the ones based on bilinear pairing [34] and vector commitments [39]. But all of them cannot allow providing non-membership proofs. Li *et al.* proposed in [91] a construction of an accumulator which supports both membership and non-membership proofs. But the proposed scheme requires a trusted accumulator manager and does not completely fit the accumulator definition given firstly by Benaloh *et al.* namely, to provide a membership proof without necessarily the need of a trusted accumulator manager. To fix this lack, Camacho *et al.* [33] proposed a strong universal accumulator which does not need a trusted accumulator manager.

All of the schemes proposed above have their security based on the discrete logarithm or the factorization problems that are not resistant to the quantum computer. This fact among others has enabled the emergency of post-quantum cryptography, like lattice-based cryptography, code-based cryptography, multivariate cryptography or isogeny-based cryptography. Then in 2015, based on lattice assumptions, a first direct construction of post-quantum cryptographic accumulator has been proposed by Jhanwar and Safavi-Naini in [85]. One year later, using Merkle-Tree, Libert *et al.* [93] presented another lattice-based cryptographic accumulator. Recently using the same technique, Derler *et al.* [54] presented a post-quantum accumulator from symmetric-key primitives, but their scheme is also static and cannot support non-membership proofs. In 2018, Ling *et al.* [96] proposed a dynamic lattice-based accumulator scheme by adding to Libert *et al.* [93] scheme an efficient update algorithm. Also in 2018, Yu *et al.* [128] proposed a lattice-based universal accumulator by adding the support of non-membership proofs to Libert *et al.* [93] scheme. In 2019, using Merkle-Tree, Nguyen *et al.* [107] proposed a first code-based cryptographic accumulator. Their proposed scheme uses random codes which need a big storage capacity and they do not provide an implementation of their scheme.

Cryptographic accumulators are very useful in many privacy-preserving technologies like anonymous credential systems, group signatures, efficient time-stamping, sanitizable signatures, ring signatures, redactable signatures, identity-based encryption, Zerocoin, *etc...* (for more detail see [53]). Group signature is a cryptographic primitive where a user can anonymously sign a message on behalf of group users. In dynamic group signature, the group manager has the possibility to add or remove users. When a group signature scheme allows to add and remove user it is called fully dynamic group signature (*FDGS*) scheme. The first post-quantum *FDGS* scheme is from lattice-based cryptography [96]. In code-based cryptography, as far we know, they are two dynamic group signature schemes [3, 14] which allow the group manager to just add users.

In this chapter, we propose a new code-based cryptographic accumulator. Like in [93], our cryptographic accumulator proceeds by computing a Merkle tree using a collision-resistant family of hash functions based on a $[2k, k]$ -code over the binary field $\mathbb{F}_2$. Moreover, this technique uses double circulant codes that provide compact keys and is securely based on the hardness of the syndrome decoding problem. Furthermore, to support the feasibility of our scheme, we give an implementation of our cryptographic accumulator which is also, as far as we know, the first direct implementation of post-quantum cryptographic accumulators. Indeed, the others are the implementations of rich cryptographic primitives that use cryptographic accumulators. Practically, for a set of 16384 elements, to have an 80 bits security level, our scheme needs only 347 bits to stock the public key, 347 bits to stock the accumulated value, 347 bits to stock

the auxiliary value and 4872 bits to stock the witness. This implementation takes less than one second to accumulate all the elements, to generate the witness of any element and to verify a witness. Since this implementation is the first direct one from post-quantum cryptographic accumulator, we cannot provide a comparison to other ones. Our construction allows to deduce a fully dynamic code-based group signature scheme which is logarithmic complexity on the group size.

The remainder of this chapter is organized as follows. In Section 7.2, we present some mandatory preliminaries and definitions to make this chapter self-contained. Section 7.3 is devoted to the description of our code-based cryptographic accumulator. We present the security analysis and the implementation of our code-based cryptographic accumulator in Section 7.4. In Section 7.5, we describe our fully dynamic code-based group signature scheme. Then we conclude in Section 7.6.

7.2 Preliminaries

In this section, we first introduce some notations and recall the definitions and security requirements for cryptographic accumulators. We also recall some necessary definitions related to the code-based cryptography.

7.2.1 Code-based Cryptography

Let $\mathbb{F}_2$ be the binary field and $\mathcal{M}_{m \times n}(\mathbb{F}_2)$ the vector space of matrices over $\mathbb{F}_2$ of m rows and n columns. We recall from [73] the Goldreich Levin Hardcore Predicate theorem (Theorem 8) and from [112] the Forking Lemma (Theorem 9).

Theorem 8 (The Goldreich Levin Hardcore Predicate Theorem). *Let λ be a security parameter. Let m, n, k be integers polynomially bounded in λ . Let $f : \mathbb{F}_2^n \times \mathbb{F}_2^m \rightarrow \mathbb{F}_2^k$ be an efficiently computable injective function. Let X and Y be a distribution of $\mathbb{F}_2^n$ and $\mathbb{F}_2^m$ respectively. Let x and y be random elements of X and Y respectively and $z = f(x, y)$. Let further r be a random element of $\mathbb{F}_2^n$ and $b = r \cdot x^\top$. Assume there exists an efficient algorithm $\mathcal{A}$ that computes b , given r and z , with probability non-negligibly better than $\frac{1}{2}$. Then there exists an efficient algorithm $\mathcal{A}'$ that computes x given z with non-negligible probability.*

Theorem 9 (The Forking Lemma). *. Let (G, Σ, V) be a generic digital signature scheme with security parameter λ and let T and T' be two positive integers. Let $\mathcal{A}$ be a probabilistic*

polynomial time Turing machine whose input only consists of public data. We denote by Q the number of queries that $\mathcal{A}$ can ask to the random oracle. Assume that, within time bound T , $\mathcal{A}$ produces, with probability $\varepsilon \geq 7Q/2^\lambda$ a valid signature $(m, \sigma_1, h, \sigma_2)$. Then there is another machine which has control over $\mathcal{A}$ and produces two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2')$ such that $h \neq h'$, in expected $T' \leq 84480TQ/\varepsilon$.

7.2.2 Zero-Knowledge Argument Systems and Stern-Like Protocols

Let $\text{RE} : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ be the function defined by Algorithm 7.1.

Algorithm 7.1 Function RE()

Require: $u \in \mathbb{F}_2^n$.

Ensure: $v \in \mathbb{F}_2^n$.

Let $v = 0^n \in \mathbb{F}_2^n$.

Split v into w sub-blocs v_i of size $\frac{n}{w}$

Split u in $\frac{n}{w \log_2(\frac{n}{w})}$ blocs of size $w \log_2(\frac{n}{w})$

for all bloc s of u **do**

 Split s in w parts $s_1, \dots, s_w$ of $\log_2(\frac{n}{w})$ bits

 Convert each bit string s_i to her corresponding integer s'_i between 1 and $\frac{n}{w}$.

 Set to 1 the bit at position s'_i in each v_i .

end for

return v

We first recall a statistically hiding and computationally binding string commitment scheme form [107]. Let γ and δ be positives integers, such that $\delta \geq \gamma$. A statistically hiding and computationally binding string commitment scheme work as follow.

- **KGen:** Randomly choose a matrices B_0 and B_1 in $\mathbb{F}_2^{\gamma \times \delta}$, and output the commitment key $B = [B_0 \parallel B_1] \in \mathbb{F}_2^{\gamma \times 2\delta}$.
- **COM:** On input a message $x \in \mathbb{F}_2^\delta$ and the commitment key B . To compute the commitment of x , we randomly choose $\rho \in \mathbb{F}_2^\delta$ and we compute the commitment $C = B_0 \cdot \text{RE}(x) \oplus B_1 \cdot \text{RE}(\rho) \in \mathbb{F}_2^\gamma$. Then we output (C, ρ) .

- Open: On input the commitment key B , the message x , the commitment C and an opening ρ , we output 1 if $C = B_0 \cdot \text{RE}(x) \oplus B_1 \cdot \text{RE}(\rho) \in \mathbb{F}_2^\gamma$ and 0 otherwise.

Now, we recall from [107] a statistical zero-knowledge arguments of knowledge which are Stern-like protocols. Let $\mathbb{V}$ be a subset of $\mathbb{F}_2^\delta$. Assume that $\mathbb{S}$ is a finite set of $\mathbb{F}_2^\delta$ such that we can associate every $\phi \in \mathbb{S}$ with a permutation Γ_ϕ of δ elements with the following conditions:

$$\begin{cases} w \in \mathbb{V} \Leftrightarrow \Gamma_\phi(w) \in \mathbb{V} \\ \text{if } w \in \mathbb{V} \text{ and } \phi \text{ is uniform in } \mathbb{S}, \text{ then } \Gamma_\phi(w) \text{ is uniform in } \mathbb{V} \end{cases} \quad (7.1)$$

In [107], the authors constructed a statistical zero-knowledge arguments of knowledge for the following abstract relation:

$$R_{\text{abstract}} = \{(M, v) \in \mathbb{F}_2^{\gamma \times \delta} \times \mathbb{F}_2^\gamma, w \in \mathbb{V} : M \cdot w = v\}. \quad (7.2)$$

To prove in Zero Knowledge that $w \in \mathbb{V}$, the prover $\mathcal{P}$ randomly chooses $\phi \in \mathbb{S}$ and let the verifier $\mathcal{V}$ check that $\Gamma_\phi(w) \in \mathbb{V}$. Furthermore, to prove in Zero Knowledge that the linear equation is hold, the prover randomly chooses a masking vector $r_w \in \mathbb{F}_2^\delta$, and convince the verifier that $M \cdot (w + r_w) = M \cdot r_w + v$. The Stern-Like interactive protocol is defined in Algorithm 7.2.

In [107], Nguyen *et al.* proved that if COM is a statistically hiding and computationally binding string commitment scheme. Then, the protocol in Algorithm 7.2 is a statistical zero-knowledge arguments of knowledge with perfect completeness, soundness error $\frac{2}{3}$.

7.2.3 Collision-Resistant Hash Function

Our proposed scheme uses collision-resistant hash functions, which we recall from [88].

Definition 20. A family of Collision-Resistant hash functions is a sequence of family of functions $\mathcal{H} = \{h : X \times Y \rightarrow Z\}$ where X, Y and Z are sets of values, with the following properties:

1. For any $h \in \mathcal{H}$, the function $h(\cdot, \cdot)$ is computable in polynomial time.
2. The probability to have a colision on $h(\cdot, \cdot)$ is less than ϵ , where ϵ is an arbitrary small number.

Algorithm 7.2 The Stern-Like Protocol for the relation $R_{abstract}$ **Require:** $(M, v) \in \mathbb{F}_2^{\gamma \times \delta} \times \mathbb{F}_2^\gamma$.**Ensure:** The value 1 or 0.

1. Let $\mathcal{P}$ randomly chooses $(r_w, \phi) \in \mathbb{F}_2^\delta \times \mathbb{S}$. Then $\mathcal{P}$ sends to $\mathcal{V}$ the commitments C_1 , C_2 and C_3 such that: $(C_1, \rho_1) = \text{COM}(\phi \oplus (M \cdot r_w \parallel 0^{(\delta-\gamma)}), M')$, $(C_2, \rho_2) = \text{COM}(\Gamma_\phi(r_w), M')$ and $(C_3, \rho_3) = \text{COM}(\Gamma_\phi(w + r_w), M')$ where $M' = [M \parallel M]$.
2. $\mathcal{V}$ sends a random value $e \in \{1, 2, 3\}$ to $\mathcal{P}$.
3. Depending on e , $\mathcal{P}$ sends R computed as follows:
 - if $e = 1$: Let $t_w = \Gamma_\phi(w)$, $t_r = \Gamma_\phi(r_w)$ and $R = (t_w, t_r, \rho_2, \rho_3)$
 - if $e = 2$: Let $\phi_2 = \phi$, $w_2 = w + r_w$ and $R = (\phi_2, w_2, \rho_1, \rho_3)$
 - if $e = 3$: Let $\phi_3 = \phi$, $w_3 = r_w$ and $R = (\phi_3, w_3, \rho_1, \rho_2)$
4. Receiving R , the verifier $\mathcal{V}$ proceeds as follows:
 - if $e = 1$: $\mathcal{V}$ verifies that if $t_w \in \mathbb{V}$, $\text{Open}(M', t_r, C_2, \rho_2) = 1$ and $\text{Open}(M', t_w + t_r, C_3, \rho_3) = 1$.
 - if $e = 2$: $\mathcal{V}$ verifies that if $\text{Open}(M', \phi_2 \oplus ((M \cdot w_2 + v) \parallel 0^{(\delta-\gamma)}), C_1, \rho_1) = 1$ and $\text{Open}(M', \Gamma_{\phi_2}(w_2), C_3, \rho_3) = 1$.
 - if $e = 3$: $\mathcal{V}$ verifies that if $\text{Open}(M', \phi_3 \oplus (M \cdot w_3 \parallel 0^{(\delta-\gamma)}), C_1, \rho_1) = 1$ and $\text{Open}(M', \Gamma_{\phi_3}(w_3), C_2, \rho_2) = 1$.

In each case, the verifier $\mathcal{V}$ outputs 1 if and only if all the conditions hold.

7.3 The Proposed Code Based Cryptographic Accumulator

This section is devoted to describe our code-based cryptographic accumulator. First, we recall from [63] the Double Circulant Regular Syndrome Decoding(DCRSD) problem and 2-Double Circulant Regular Null Syndrom Decoding(2-DCRNSD) problem and we introduce the Decisional Double Circulant Regular Syndrome Decoding(DDCRSD) problem which will be useful for the security proofs. Second, we construct a family of collision-resistant hash functions. And third, we use these hash functions to propose a code-based cryptographic accumulator.

7.3.1 A Code Based Difficult Problems

Problem 6 (Double Circulant Regular Syndrome Decoding(DCRSD) problem[12]). *Given an integer w , a vector $s \in \mathbb{F}_2^k$ and a double circulant matrix $H \in \mathcal{M}_{k \times n}(\mathbb{F}_2)$ split into w sub-blocs H_i of size $k \times \frac{n}{w}$, find w columns of H , one per block H_i , such that s is the sum of the w columns.*

Problem 7 (2-Double Circulant Regular Null Syndrom Decoding(2-DCRNSD) problem[12]). *Given a double circulant matrix $H \in \mathcal{M}_{k \times n}(\mathbb{F}_2)$ split into w sub-blocs H_i of size $k \times \frac{n}{w}$, find $2w'$ columns (with $0 < w' \leq w$) of H , 0 or 2 per block H_i , such that the sum of the $2w'$ columns is null.*

In [12], Augot *et al.* showed that DCRSD and 2-DCRNSD problems are hard. Now we introduce the Decisional Double Circulant Regular Syndrome Decoding(DDCRSD) problem.

Problem 8 (Decisional Double Circulant Regular Syndrome Decoding(DDCRSD) problem). *Given a pair $(A, v) \in \mathcal{M}_{k \times n}(\mathbb{F}_2) \times \mathbb{F}_2^k$, distinguish whether (A, v) is a uniformly random pair over $\mathcal{M}_{k \times n}(\mathbb{F}_2) \times \mathbb{F}_2^k$ or it is obtained by randomly choosing $A \in \mathcal{M}_{k \times n}(\mathbb{F}_2)$ and outputting (A, v) , such that after splitting A into w sub-blocs A_i of size $k \times \frac{n}{w}$ and choosing w columns of A , one per block A_i , v is the sum of the w columns.*

Proposition 6. *The DDCRSD problem (Problem 8) is as hard as the DCRSD problem (Problem 6).*

Proof. In this proof, we make a probabilistic reduction of DDCRSD problem (Problem 8) to DCRSD problem (Problem 6). Compute a couple (A, v) , such that after splitting A into w sub-blocs A_i of size $k \times \frac{n}{w}$ and choosing w columns of A , one per block A_i , v is the sum of the w columns is equivalent to choose a random vector u of $\mathbb{F}_2^n$ such that when u split u in w blocs, each bloc contains only one bit set to 1 and compute $v = A \cdot u^\top$. We assume that there exist a PPT algorithm $\mathcal{D}$ that takes as input an instance of DDCRSD problem and output 0, with a non-negligible advantage ε , if the couple (A, v) is a uniformly random and 1 otherwise. It means that

$$Pr[\mathcal{D}(A, v) = 1] - Pr[\mathcal{D}(A, A \cdot u^\top) = 1] \geq \varepsilon. \quad (7.3)$$

For $r \in \mathbb{F}_2^n$, let $\mathcal{A}'$ (Algorithm 7.3 below) be an algorithm that computes the Goldreich-Levin hardcore-bit $r \cdot u^\top$ of u given $(A, A \cdot u^\top)$ with advantage $\frac{\varepsilon}{2}$. (See Theorem 8)

Now let $(A, A \cdot u^\top)$ and r be the input of $\mathcal{A}'$. Since $A = B + v^\top \cdot r$ it follows that

$$y = A \cdot u^\top = B \cdot u^\top + v^\top \cdot r \cdot u^\top = B \cdot u^\top + v^\top (r \cdot u^\top).$$

On one hand, if $r \cdot u^\top = 0$, then (B, y) has the distribution $(B, B \cdot u)$. On the other hand, if $r \cdot u^\top = 1$, then $y = B \cdot u^\top + v^\top$. As v is a random vector, y is also random vector. Thus (B, y) has the distribution (B, v') , for a random vector v' . Let $\text{Succ}(\mathcal{A}')$ be the probability of $\mathcal{A}'$ success. We get:

$$\begin{aligned} \text{Succ}(\mathcal{A}') &= \Pr[\mathcal{A}'((A, A \cdot u^\top), r) = r \cdot u^\top] \\ &= \frac{1}{2} \Pr[\Pr[\mathcal{A}'((A, A \cdot u^\top), r) = 0 \mid r \cdot u^\top = 0]] \\ &\quad + \frac{1}{2} \Pr[\Pr[\mathcal{A}'((A, A \cdot u^\top), r) = 1 \mid r \cdot u^\top = 1]] \\ &= \frac{1}{2} \Pr[\Pr[\mathcal{A}'((B, B \cdot u^\top), r) = 1]] + \frac{1}{2} \Pr[\Pr[\mathcal{A}'((B, v'), r) = 1]] \\ &= \frac{1}{2} + \frac{1}{2} \left(\Pr[\mathcal{D}(A, v) = 1] - \Pr[\mathcal{D}(A, A \cdot u^\top) = 1] \right) \end{aligned}$$

then from (7.3) we get $\text{Succ}(\mathcal{A}') \geq \frac{1}{2} + \frac{\varepsilon}{2}$.

It means that $\mathcal{A}'$ computes the Goldreich-Levin hardcore bit $r \cdot u$ with probability non-negligibly better than $\frac{1}{2}$. By the Goldreich Levin theorem (Theorem 8) this algorithm $\mathcal{A}'$ can be used to construct an algorithm that solves the Double Circulant Regular Syndrome Decoding problem (Problem 6). $\square$

Algorithm 7.3 Adversary $\mathcal{A}'$

Require: $(A, y) \in \mathbb{F}_2^{k \times n} \times \mathbb{F}_2^k$, $r \in \mathbb{F}_2^n$

Ensure: b the Goldreich-Levin hardcore-bit

choose a random $v \in \mathbb{F}_2^k$

compute $B = A - v^\top \cdot r$

$b = \mathcal{D}(B, y)$, where $\mathcal{D}$ is a in (7.3)

return b

7.3.2 A Family of Code-Based Collision-Resistant Hash Functions

In 2003, Augot *et al.* [13] presented a first family of code-based hash function called Syndrome Based (SB) hash function using random codes. To improve SB that has been proved to be slow due to its compression function that uses very big integers, Augot *et al.* [12] presented Fast Syndrome Based (FSB) algorithm. FSB has a particularity to use a compression function with

regular coding that uses little numbers. In 2008, Finiasz *et al.* [63] proposed to use quasi-cyclic codes instead of random codes which improve the speed of FSB. The family of a hash function (7.4) presented in this chapter is mainly based on FSB. Instead of random codes used in [12], our family of hash functions uses an $[n, k]$ random double circulant codes where k is an integer > 0 . Let n, k and t be non negative integers such that $n = 2k$. The function family $\mathcal{H}$ mapping $\mathbb{F}_2^k \times \mathbb{F}_2^k$ to $\mathbb{F}_2^k$ is defined as

$$\mathcal{H} = \{h_H \mid H \in \mathcal{M}_{k \times n}(\mathbb{F}_2)\} \quad (7.4)$$

where H is a parity check matrix of an $[n, k]$ code C whose correction capacity is t and where h_H is computed by Algorithm 7.4.

Algorithm 7.4 Hash function h_H

Require: $(x, y) \in \mathbb{F}_2^k \times \mathbb{F}_2^k, H \in \mathcal{M}_{k \times n}(\mathbb{F}_2)$

Ensure: h the hash of $x \parallel y$. Where $\parallel$ denote the concatenation.

$u \leftarrow x \parallel y$

$h \leftarrow H \cdot \text{RE}(u)$, RE is as in Algorithm 7.1.

return h

We study the security of the family of hash functions in Proposition 7.

Proposition 7. *The family of hash function defined by Algorithm 7.4 is one way and collision resistant.*

Proof. We assume that there exist an algorithm $\mathcal{A}$ that can inverse the hash function h_H presented by Algorithm 7.4. It means that, given a double circulant matrix $H \in \mathcal{M}_{k \times n}(\mathbb{F}_2)$ and $x \in \mathbb{F}_2^k$, $\mathcal{A}$ can find a word $y \in \mathbb{F}_2^n$ of weight w that have only one non zero position per block of size $\frac{n}{w}$ such that $x = h_H(y)$. In other words $\mathcal{A}$ can solve the Double Circulant Regular Syndrome Decoding problem (Problem 6) which is a difficult problem.

Now, we assume that there exist an algorithm $\mathcal{A}'$ that can find a collision on the hash function h_H . It means that $\mathcal{A}'$ can find two different words $x_1 \in \mathbb{F}_2^n$ and $x_2 \in \mathbb{F}_2^n$ of weight w that are only one non zero position per block of $\frac{n}{w}$ such that $Hx_1 = Hx_2$ then, $H(x_1 + x_2) = 0$. Let $x = x_1 + x_2$. It is clear that x is a word of weight equal or less than $2w$ which contains either 0 or 2 non-zero positions in each block of size $\frac{n}{w}$. In other words, find a collision on the hash function h_H is equivalent to solve 2-Double Circulant Regular Null Syndrome Decoding problem (Problem 7) which is a difficult problem. $\square$

7.3.3 The Proposed Code-based Accumulator

Let ℓ be an integer ≥ 0 . Our code-based accumulator uses a Merkle tree with $N = 2^\ell$ leaves. This accumulator scheme is based on the family of code-based collision-resistant hash functions presented above. In the following, we recall from [83], the well-known definition of Merkle Tree.

Definition 21. *A binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child.*

The length of the longest path from the root to a leaf is said height, where the length between two leafs is the number of node between these two leafs. Also, we say that a binary tree of height ℓ is complete if it has 2^ℓ leaves and $2^\ell - 1$ interior nodes.

Let h be a one-way hash function and Φ a function which maps the set of nodes of arbitrary length to the set of k -length strings: $n \mapsto \Phi(n) \in \{0, 1\}^k$. A Merkle tree is a complete binary tree equipped with two functions h and Φ . For two children nodes n_{left} and n_{right} , of any interior node n_{parent} , the function Φ satisfies:

$$\Phi(n_{parent}) = \Phi(n_{left} \parallel n_{right})$$

where $\parallel$ is the concatenation symbol.

As defined by the cryptographic accumulator model in [53], our accumulator is composed by Algorithms 7.5 to 7.8 hereafter.

Gen(1^λ): This algorithm (Algorithm 7.5 below) outputs the global parameters (n, k, t) needed to generate our double circulant code C and the public parameter $p_k = H \in \mathcal{M}_{k \times n}(\mathbb{F}_2)$ which is a parity check matrix of C .

Algorithm 7.5 Gen() Algorithm

Require: 1^λ a security parameter

Ensure: $(k, n, t), p_k$

pick $(k, n, t) \in \mathbb{N}^3$, such that there exist an (n, k, t) double circulant code and Problems 6 and 7 get a security level of λ bits where $n = 2k$.

Pick randomly $H \in \mathcal{M}_{k \times n}(\mathbb{F}_2)$, a parity check matrix of an (n, k, t) double circulant code;

$p_k \leftarrow H$.

Eval($\mathbf{p}_k, \mathcal{X}$): Using a Merkle tree, Algorithm 7.6 takes as input a set of values $\mathcal{X} = \{x_0, \dots, x_{N-1}\}$ to accumulate in a value $A_{\mathcal{X}}$ and auxiliary value ξ . For each $j \in [0, N-1]$, let $(j_1, \dots, j_\ell)$ be the binary representation of j on ℓ bits and write $x_j = y_{j_1, \dots, j_\ell}$. We design Algorithm 7.6 from the tree of depth ℓ based on N leaves $y_{0, \dots, 0}, \dots, y_{1, \dots, 1}$.

Algorithm 7.6 Eval() Algorithm

Require: $\mathbf{p}_k, \mathcal{X}$

Ensure: $A_{\mathcal{X}}, \xi$

set ℓ as the number of bits of $N-1$

randomly chosen $u \in \mathbb{F}_2^k$

set $y_{1,1, \dots, 1}$ and ξ to u

for $i = \ell, \dots, 0$ **do**

$y_{j_1, \dots, j_i} = h_H(y_{j_1, \dots, j_i, 0}, y_{j_1, \dots, j_i, 1})$, where $(j_1, \dots, j_i) \in \{0, 1\}^i$

if $i = 0$ **then**

$A_{\mathcal{X}} \leftarrow h_H(y_0, y_1)$

end if

end for

WitCreate($\mathbf{p}_k, \mathcal{X}, A_{\mathcal{X}}, \xi, x$): Algorithm 7.7 uses the public key $\mathbf{p}_k$ and the Eval() algorithm to output the witness W_x associate to x .

Algorithm 7.7 WitCreate() Algorithm

Require: $\mathbf{p}_k, \mathcal{X}, \xi, A_{\mathcal{X}}, x$

Ensure: the witness W_x of x

if $x \notin \mathcal{X}$ **then**

return error

else

set $y_{1,1, \dots, 1}$ to ξ

find the relative position j of x in $\mathcal{X}$

write j in binary form: $(j_1, \dots, j_\ell) \in \{0, 1\}^\ell$

$W_x = ((j_1, \dots, j_\ell), (y_{j_1, \dots, j_{\ell-1}, \bar{j}_\ell}, \dots, y_{j_1, \bar{j}_2}, y_{\bar{j}_1}))$ where $y_{j_1, \dots, j_{\ell-1}, \bar{j}_\ell}, \dots, y_{j_1, \bar{j}_2}, y_{\bar{j}_1}$ are computed as in Algorithm 7.6, and $\bar{j}_i = j_i \oplus 1$, where $\oplus$ is the bitwise operation.

end if

TVerify($\mathbf{p}_k, A_{\mathcal{X}}, W_x, x$) defined in Algorithm 7.8, proves using a recursive function that W_x is or not a generated witness of x .

At the beginning, the system uses Gen() Algorithm 7.5 to output and publish the public parameters $(k, n, t) \in \mathbb{N}^3$ and the public key $H \in \mathcal{M}_{k \times n}(\mathbb{F}_2)$. Thereafter the system randomly

Algorithm 7.8 TVerify() Algorithm

Require: $p_k, A_{\mathcal{X}}, x, W_x$
Ensure: true or false
transcript W_x to $((j_1, \dots, j_\ell), (w_\ell, \dots, w_1))$
 $v_\ell \leftarrow x$
for $i = \ell - 1$ **to** 0 **do**
 if $j_{i+1} = 0$ **then**
 $v_i = h_H(v_{i+1}, w_{i+1})$
 else
 $v_i = h_H(w_{i+1}, v_{i+1})$
 end if
end for
if $v_0 = A_{\mathcal{X}}$ **then**
 return true
else
 return false
end if

generates a set of $N - 1$ values $\mathcal{X} = \{x_0, \dots, x_{N-2}\}$ where $x_i \in \mathbb{F}_2^k$. After all the operations above, the system computes and publish the accumulated value $A_{\mathcal{X}}$ and the auxiliary value ξ by providing the public key $p_k = H$ and the set $\mathcal{X}$ to Eval() Algorithm 7.6. Now everyone can compute the witness W_x of any element $x \in \mathcal{X}$ by providing the public key p_k , the auxiliary value ξ and the element x to WitCreate() Algorithm 7.7. At the end, everyone can also verify the membership of x to $\mathcal{X}$ by providing the public key p_k ; the accumulated value $A_{\mathcal{X}}$, the element x and the witness W_x to TVerify() Algorithm 7.8.

7.3.4 Example

We present hereafter an example of our scheme to well understand the mechanism of each algorithm presented above.

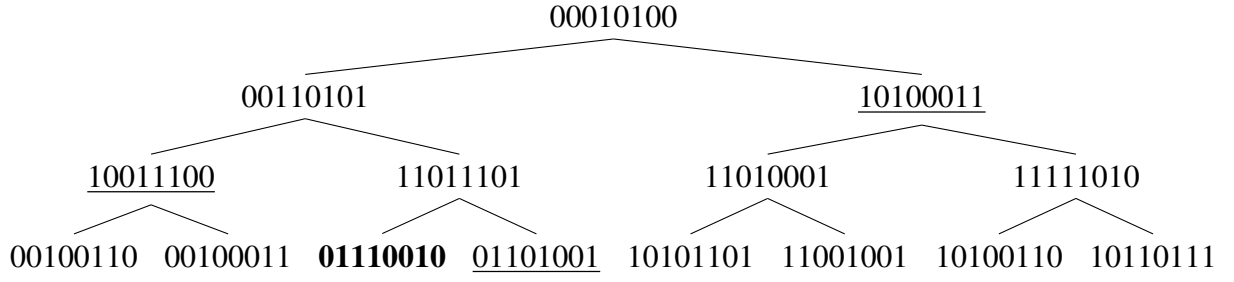


Fig. 7.1: Merkel Tree Associate to the set $\mathcal{X}$

Let C be a $[16, 8]$ -code which a parity check matrix is H is generated by $\text{Gen}()$ function (Algorithm 7.5).

$$H = \left(\begin{array}{c|c} I_8 & \begin{matrix} 10111001 \\ 11011100 \\ 01101110 \\ 00110111 \\ 10011011 \\ 11001101 \\ 11100110 \\ 01110011 \end{matrix} \end{array} \right) \quad (7.5)$$

In this example we take the following set

$$\mathcal{X} = \{00100011, 01110010, 01101001, 10101101, 11001001, 10100110, 10110111\}$$

of $2^3 - 1 = 7$ elements. The execution of $\text{Eval}()$ function (Algorithm 7.6) output two values: the accumulated value $A_{\mathcal{X}} = 00010100$ and the auxiliary value $\xi = 00100110$. In Figure 7.1 we give the Merkle tree associate to this example.

The root of the tree is the accumulated value $A_{\mathcal{X}} = 00010100$ and the $2^3 = 8$ leaves are the auxiliary value $\xi = 00100110$ and the elements of the set $\mathcal{X}$. The first step to generate the witness of the element $x = \mathbf{01110010}$ which is in bold on Figure 7.1 is to get its relative position on the set $\mathcal{X}$ which is 2 and 010 in binary representation. The second step is to use $\text{WitCreate}()$ function (Algorithm 7.7) to get the underlined elements on Figure 7.1. Then the witness W_x of $x = \mathbf{01110010}$ is $W_x = \{010, 01101001, 10011100, 10100011\}$. The only way to check the membership of $x = \mathbf{01110010}$ is to use $\text{TVerify}()$ function (Algorithm 7.8) which should output True.

7.4 Security and Implementation of the Proposed Cryptographic Accumulator

In this section, we focus on the security and the implementation of the proposed scheme.

Notation: In the following, $\mathcal{O}^E$ represents an oracle for the algorithm Eval() and an adversary $\mathcal{A}$ that have access to an oracle $\mathcal{O}$ is noted $\mathcal{A}^{\mathcal{O}}$. An adversary is allowed to query them an arbitrary number of times. The oracle $\mathcal{O}^W$ allows the adversary to obtain membership witnesses for members. In Collision Freeness Experiment (Algorithm 2.10), $A_{\mathcal{X}}$ is the execution result of Eval($p_k, \mathcal{X}$) and the adversary $\mathcal{A}$ has oracle access to $\mathcal{O}^E$ and $\mathcal{O}^W$. In Indistinguishability Experiment (Algorithm 2.11), $\mathcal{X}_0$ and $\mathcal{X}_1$ are two distinct subsets of the accumulation domain and $state$ is additional informations on $\mathcal{X}_0$ and $\mathcal{X}_1$. We define $1^\lambda = \overbrace{11 \dots 1}^{\lambda \text{ times}}$ as a string of ones λ times where λ is a security parameter and $Pr[A]$ is the probability to have the event A realized.

7.4.1 Security of the Proposed Cryptographic Accumulator

The security analysis of the proposed scheme consists in proving the correctness and the Collision Freeness. Additionally to these properties, we also prove the Indistinguishability of our proposal.

Correctness: To ensure the correctness of our scheme, we show that the execution of TVerify() Algorithm 7.8 will always succeed if the key is honestly generated and accumulators and witnesses are correctly computed.

Lemma 1. *If all Gen(), Eval() and WitCreate() algorithms are correctly computed, then the execution of TVerify() algorithm outputs true.*

Proof. To verify the membership of an element with a witness

$$W_x = ((j_1, \dots, j_\ell), (y_{j_1, \dots, j_{\ell-1}, \bar{j}_\ell}, \dots, y_{j_1, \bar{j}_2}, y_{\bar{j}_1})),$$

the TVerifyAlgorithm 7.8 computes at the end for $i = 0$:

$$\begin{aligned} v_0 &= h_H(y_1, y_{\bar{j}_1}), \text{ if } j_1 = 0 \text{ and} \\ v_0 &= h_H(y_1, y_{\bar{j}_1}) \text{ otherwise} \end{aligned}$$

which means that $v_0 = h_H(y_0, y_1) = A_{\mathcal{X}}$. □

Collision Freeness: To ensure the collision freeness properties, we show that an adversary cannot find a witness for a non-accumulated value.

Lemma 2. *If the family of hash function defined by Algorithm 7.4 is resistant to collisions then our cryptographic accumulator is collision freeness according to Definition 15.*

Proof. Let $\mathcal{A}$ be an adversary against the collision freeness of our cryptographic accumulator scheme with a success probability ε where $\varepsilon \in [0; 1]$. We construct an adversary $\mathcal{B}$ that can find a collision on the family of hash function defined by Algorithm 7.4 with a success probability related to ε .

The adversary $\mathcal{B}$ receives a hash function instance $\tilde{H} \in \mathcal{M}_{k \times n}(\mathbb{F}_2)$. The goal is to find vectors (x_1, y_1) and (x_2, y_2) in $\mathbb{F}_2^k \times \mathbb{F}_2^k$ such that $(x_1, y_1) \neq (x_2, y_2)$ and $h_{\tilde{H}}(x_1, y_1) = h_{\tilde{H}}(x_2, y_2)$. Then $\mathcal{B}$ proceeds as follows:

1. Pick a random set $\mathcal{X} = \{x_0, \dots, x_{N-2}\}$ and a random $x \in \mathbb{F}_2^k$ such that $x \notin \mathcal{X}$ where each $x_i \in \mathbb{F}_2^k$.
2. Set H to $\tilde{H}$.
3. Send $p_k = H, \mathcal{X}$ and x to $\mathcal{A}$.

The adversary $\mathcal{B}$ then handles the queries from $\mathcal{A}$ as follows:

- $\mathcal{O}^{\text{Eval}()}:$ Call the original Eval() function. If $x \in \mathcal{X}$ then $\mathcal{B}$ aborts.
- $\mathcal{O}^{\text{WitCreate}()}:$
 - If $x \in \mathcal{X}$: the adversary $\mathcal{B}$ compute honestly the witness W_x of x .

- If $x \notin \mathcal{X}$: the adversary $\mathcal{B}$ return a simulate witness W_x of x .

Then, the adversary $\mathcal{A}$ output a forged witness W_x for $x \notin \mathcal{X}$ where

$$W_x = ((j_1, \dots, j_\ell), (y_{j_1, \dots, j_{\ell-1}, \bar{j}_\ell}, \dots, y_{j_1, \bar{j}_2}, y_{\bar{j}_1}))$$

Because $\mathcal{A}$ can break the collision freeness then $\text{TVerify}(\mathbf{p}_k, A_{\mathcal{X}}, x, W_x) = \text{true}$. It means that, at some point of the execution of $\text{TVerify}()$ Algorithm we hold the following equation:

$$h_H(y_0, h_H(y_1, h_H(\dots, h_H(y_{j_1, \dots, j_{\ell-1}, \bar{j}_\ell}, x)))) = A_{\mathcal{X}}.$$

Then, there exist values $v_1 \in \mathcal{X}$ and $v_2 \in \mathcal{X}$ such that

$$H(y_{j_1, \dots, j_{\ell-1}, \bar{j}_\ell}, x) = h_H(v_1, v_2).$$

In other words, if $\mathcal{A}$ can break the collision freeness of our cryptographic accumulator with a probability ε then the adversary $\mathcal{B}$ can find a collision on the family of hash functions defined by Algorithm 7.4 with a probability close to ε . $\square$

Theorem 10. *The proposed cryptographic accumulator is secure based on the collision resistant property of the family of hash function defined by Algorithm 7.4.*

Proof. A cryptographic accumulator is secure if it is correct and collision-free. Whereas, from Lemma 1 our cryptographic accumulator is correct and from Lemma 2 we have the collision freeness of our scheme. Then the proposed cryptographic accumulator is secure. $\square$

Indistinguishability: To ensure the Indistinguishability of the proposed cryptographic accumulator, we prove that from two sets and an accumulated value, an adversary cannot guess what set was accumulated.

Theorem 11. *The proposed cryptographic accumulator defined by Algorithms 7.5 to 7.8 is Indistinguishable based on the hardness of the DDCRS Problem (Problem 8).*

Proof. Let $\mathcal{A}$ be an adversary who can break the indistinguishability of our cryptographic accumulator with a probability of $\frac{1}{2} + \varepsilon$ where $\varepsilon \in [0; \frac{1}{2}]$. We prove that this probability is close to $\frac{1}{2}$ assuming the hardness of the DDCRS problem (Problem 8).

We consider the Experiment $\mathcal{Exp}^b$ which depends on a bit b and the Experiment $\mathcal{Exp}$ as follows:

Experiment $\mathcal{Exp}^b$. We describe this experiment in two phases. The first is the Challenge phase and the second is the Guess phase.

Challenge phase:

1. The adversary $\mathcal{A}$ receives the public key p_k and two different sets $\mathcal{X}_0$ and $\mathcal{X}_1$.
2. Depending on the randomly chosen bit $b \in \{0, 1\}$, the adversary $\mathcal{A}$ receives $A_{\mathcal{X}_b}$ where $A_{\mathcal{X}_b} = \text{Eval}(p_k, \mathcal{X}_b)$.
3. The adversary $\mathcal{A}$ can make $\text{Eval}()$ and $\text{WitCreate}()$ Oracles at will, since he/she does not use the sets $\mathcal{X}_0$ or $\mathcal{X}_1$.

Guess phase: The adversary $\mathcal{A}$ then outputs b with probability $\frac{1}{2} + \epsilon$.

Experiment $\mathcal{Exp}$. In this experiment we introduce the following modification on $\mathcal{Exp}^b$: in the challenge phase we simulate the $\text{Eval}()$ function to output a random value $r \in \mathbb{F}_2^k$.

At this step, we show that if $\mathcal{A}$ can distinguish $\mathcal{Exp}^b$ and $\mathcal{Exp}$ then there exist a distinguisher $\mathcal{D}$ which can solve the DDCRSD problem. Given a vector $r^* \in \mathbb{F}_2^k$ the goal is to guess if r^* was randomly generated or was generated by a hash function h_H . The distinguisher $\mathcal{D}$ receives the challenge r^* and can interact with $\mathcal{A}$ as follow:

Setup: Generates the sets $\mathcal{X}_0$ and $\mathcal{X}_1$, execute the $\text{Gen}()$ algorithm (Algorithm 7.5) and send $\mathcal{X}_0$, $\mathcal{X}_1$ and p_k to $\mathcal{A}$. The Challenge is to set $A_{\mathcal{X}_b} = r^*$ and output $A_{\mathcal{X}_b}$.

Finally, on the one hand it is clear that if $A_{\mathcal{X}_b}$ is generated by the family of hash function defined by (7.4) then the view of $\mathcal{A}$ after its interaction with $\mathcal{D}$ is close to $\mathcal{Exp}^b$. On the other hand, if $A_{\mathcal{X}_b}$ is a random vector then the point of view of $\mathcal{A}$ is close to $\mathcal{Exp}$. We stress that $\mathcal{Exp}$ does not lie to the challenge bit b , thus the advantage of $\mathcal{A}$ in $\mathcal{Exp}$ is null. Therefore, if $\mathcal{A}$ can guess which of $\mathcal{Exp}^b$ or $\mathcal{Exp}$ is interacting with the challenger with a probability larger than $\frac{1}{2}$ then $\mathcal{D}$ can use $\mathcal{A}$'s guess to solve the instance r^* of DDCRSD problem with the same probability. $\square$

7.4.2 Implementation

In this part, we evaluate the performances of our cryptographic accumulator in three parts. The first one is about the memory space needed and the second one consists in the computational coast. In the third part, we give the implementation details and results. There are no numerical results in the literature, as far as we know, for post-quantum based cryptographic accumulators to compare with.

Memory Storage. Our cryptographic accumulator uses a double circulant codes because of his structure also double circulant codes allows us to get a smaller key size. In what follows, we give the space needed to store the different elements of our cryptographic accumulator. To stock the public key $p_k = H \in \mathcal{M}_{k \times 2k}(\mathbb{F}_2)$, where $H = (I_k \mid A)$ is a parity check matrix of a $[2k, k]$ code where $A \in \mathcal{M}_{k \times k}(\mathbb{F}_2)$ such that A is a circulant matrix and $I_k \in \mathcal{M}_{k \times k}(\mathbb{F}_2)$ is the $(k \times k)$ identity matrix, we just need the first line of A . Then we need k bits to stock p_k . Since $A_{\mathcal{X}} \in \mathbb{F}_2^k$ his storage needs k bits and since $\xi \in \mathbb{F}_2^k$ his storage needs k bits. The witness is a set of $\ell = \log_2(N)$ and ℓ vectors of $\mathbb{F}_2^k$ where N is the number of vector to accumulate. Then we need $\ell(k+1)$ bits to store W_x .

Computational cost. Let μ_k and σ_k be respectively one multiplication of two polynomials of $\frac{\mathbb{F}_2[X]}{X^k+1}$ and the bitwise operation between two vectors of $\mathbb{F}_2^k$. We now give for each algorithm of our cryptographic accumulator the computational cost:

- Eval: the complete execution of the Eval() Algorithm uses $\frac{\ell}{2}\sigma_k$ and $\ell\mu_k$ operations.
- WitCreate: this algorithm uses the same number of operations as the Eval() Algorithm . Then WitCreate() Algorithm uses $\frac{\ell}{2}\sigma_k$ and $\ell\mu_k$ operations.
- TVerify: to verify if a value is or not a generated witness of an other value, we need to do $\frac{3\ell}{2}\sigma_k$ and $\ell\mu_k$ operations.

Implementation. Now, we present the implementation results of the proposed code-based cryptographic accumulator to prove its practicability. To make this test, we have used a computer running at 2.6 GHz CPU with 8 GB of RAM. For error-correcting codes generating and all algebraic operations, we use the FlexyProvider library [126] implemented in Java language. To get 80-bit security, we choose $n = 694$, $k = 347$ and $t = 76$ as used in [10].

N	pk size	$A_{\mathcal{X}}$ size	ξ size	W_x size	Eval(ms)	WitCreate(ms)	Verify(ms)
64	347	347	347	2088	1.705	1.981	0.190
128	347	347	347	2436	3.522	3.986	0.257
1024	347	347	347	3480	29.901	26.465	0.329
16384	347	347	347	4872	434.343	363.928	0.448

Table 7.1: The algorithms execution time and memory storage needed (in Bit)

In Table 7.1, we present the results of our implementation, which consist of the execution time in milliseconds (ms) of different algorithms of our proposed cryptographic accumulator. Additional to implementations results, Table 7.1 shows public key, the accumulated value, the auxiliary value and the witness size with respect to various numbers of group values to accumulate. It is easy to see that the size of the public key, just like those of the accumulator value and the auxiliary value does not depend on the group size. Also, the size of the witness is logarithmic in the group size.

7.5 The Proposed Fully Dynamic Group Signature scheme

In this section, we propose a code-based fully dynamic group signature scheme which is trap-door free. Our construction uses the Merkel tree cryptographic accumulator presented in Section 7.3 to certify users. Before defining our Fully Dynamic Group Signature scheme, we need to recall the logarithmic-size arguments of set membership from [107] and equip our Merkel tree cryptographic accumulator to an update algorithm. By this update algorithm it will be possible to efficiency add and remove users from our *FDGS*.

7.5.1 Logarithmic-Size Arguments of Set Membership

Here, given a uniformly random matrix $H \in \mathcal{M}_{k \times 2k}(\mathbb{F}_2)$ and the accumulated value $u \in \mathbb{F}_2^k$, the prover $\mathcal{P}$ have to convince the verifier $\mathcal{V}$ in zero-knowledge that $\mathcal{P}$ possesses a value d and

the associate valid witness w . To do this, Nguyen *et al.* [107] have reduced the relation R_{acc} , as follows, to the relation $R_{abstract}$ defined in (7.2).

$$R_{acc} = \left\{ ((H, u) \in \mathcal{M}_{k \times 2k}(\mathbb{F}_2) \times \mathbb{F}_2^k; d \in \mathbb{F}_2^k, w \in \mathbb{F}_2^\ell \times (\mathbb{F}_2^k)^\ell) : \text{TVerify}(H, u, w, d) = 1 \right\} \quad (7.6)$$

The construction of this zero-knowledge argument of knowledge need the following permuting techniques:

- For vector $b = (b_1, \dots, b_n) \in \mathbb{F}_2^n$, where n is a positive integer, $Encode(b)$ denote the vector $b = (\bar{b}_1, b_1, \dots, \bar{b}_1, b_n) \in \mathbb{F}_2^{2n}$.
- Let $I_n^* \in \mathcal{M}_{n \times 2n}(\mathbb{F}_2)$ be an extension of the identity matrix I_n , obtained by inserting a zero-column 0^n right before each columns of I_n .
- For $t = (t_1, \dots, t_n) \in \mathbb{F}_2^n$, the permutation F_t' transform the vector

$$w = (w_{1,0}, w_{1,1}, \dots, w_{n,0}, w_{n,1})^\top \in \mathbb{F}_2^{2n}$$

into:

$$F_t'(w) = (w_{1,t_1}, w_{1,\bar{t}_1}, \dots, w_{n,t_n}, w_{n,\bar{t}_n})^\top.$$

- For $b \in \{0, 1\}$ and $v \in \mathbb{F}_2^n$, $Ext(b, v)$ denote the vector $\begin{pmatrix} \bar{b} \cdot v \\ b \cdot v \end{pmatrix}$.
- Let c be a positive integer. For every $s = (s_1, \dots, s_c) \in \mathbb{F}_2^c$, define the permutation E_s that transforms vector $x = (x_{0,\dots,0}, \dots, x_{i_1,\dots,i_c}, \dots, x_{1,\dots,1}) \in \mathbb{F}_2^{2^c}$ into vector

$$E_s(x) = (x'_{0,\dots,0}, \dots, x'_{1,\dots,1})$$

where for each $(i_1, \dots, i_c) \in \mathbb{F}_2^c$ we have $x_{i_1,\dots,i_c} = x'_{i_1 \oplus s_1, \dots, i_c \oplus s_c}$.

- For t in $(t_1 \parallel \dots \parallel t_{n/c})$ consisting of n/c blocks of length c , define the permutation E_t' that transform $y = (y_1 \parallel \dots \parallel y_{n/c}) \in \mathbb{F}_2^{2^{c \cdot n/c}}$ to $E_t'(y) = (E_{t_1}(y_1) \parallel \dots \parallel E_{t_{n/c}}(y_{n/c}))$.

- For a bit $e \in \{0, 1\}$, for $t \in \mathbb{F}_2^k$, define the permutation $\psi_{e,t}$ that acts on $z = \begin{pmatrix} z_0 \\ z_1 \end{pmatrix} \in \mathbb{F}_2^{2k}$, where $z_0, z_1 \in \mathbb{F}_2^k$ as follows. It transforms z to $\psi_{e,t}(z) = \begin{pmatrix} E'_t(z_e) \\ E'_t(z_{\bar{e}}) \end{pmatrix}$, where $\bar{e} = e \oplus 1$.

Now we can examine the equations associated to the relation R_{acc} defined in (7.6). By focusing on the TVerify Algorithm 7.8 that computes the path $v_\ell = d, v_{\ell-1}, \dots, v_1, v_0$ as follows. For $i \in \{\ell-1, \dots, 1, 0\}$:

$$v_i = \begin{cases} h_H(v_{i+1}, w_{i+1}), & \text{if } j_{i+1} = 0; \\ h_H(w_{i+1}, v_{i+1}), & \text{if } j_{i+1} = 1. \end{cases} \quad (7.7)$$

then:

$$v_i = \begin{cases} H \cdot y_1 + H \cdot z_1 = u; \\ H \cdot y_2 + H \cdot z_2 + I_n^* \cdot x_1 = 0. \\ \dots \\ H \cdot y_\ell + H \cdot z_\ell + I_n^* \cdot x_{\ell-1} = 0. \end{cases} \quad (7.8)$$

where $(x_i)_{i \in [0, \ell-1]} = \text{Encode}(v_i)$ and for $i \in [0, \ell]$

$$\begin{cases} y_i = \text{Ext}(j_i, \text{RE}(v_i)) \in \mathbb{F}_2^{2k} \\ z_i = \text{Ext}(\bar{j}_i, \text{RE}(w_i)) \in \mathbb{F}_2^{2k}. \end{cases} \quad (7.9)$$

Now, using the linear algebra, the equation (7.8) can be unifying in the form $M \cdot w = v$ where $M \in \mathcal{M}_{\ell k \times \delta}(\mathbb{F}_2)$, $v \in \mathbb{F}_2^{\ell k}$ are public and $w \in \mathbb{F}_2^\delta$ is secret with $\delta = 2k(3\ell - 1)$ and

$$w = (y_1 \parallel \dots \parallel y_\ell \parallel z_1 \parallel \dots \parallel z_\ell \parallel x_1 \parallel \dots \parallel x_{\ell-1}) \quad (7.10)$$

Now, let focus on the set $\mathbb{V}$ containing our secret w , the set $\mathbb{S}$ and the permutations $\{\Gamma_\phi : \phi \in \mathbb{S}\}$ such that the conditions in (7.1) is hold. Let $\mathbb{V}$ be the set of all

$$w' = (y'_1 \parallel \dots \parallel y'_\ell \parallel z'_1 \parallel \dots \parallel z'_\ell \parallel x'_1 \parallel \dots \parallel x'_{\ell-1}) \in \mathbb{F}_2^\delta$$

satisfying the following conditions.

- For $i \in [0, \ell]$, there exist $v'_i, w'_i \in \mathbb{F}_2^k, j'_i \in \{0, 1\}$ such

$$y'_i = \text{Ext}(j'_i, \text{RE}(v'_i)) \in \mathbb{F}_2^{2k}, \text{ and } z'_i = \text{Ext}(\overline{j'_i}, \text{RE}(w'_i)) \in \mathbb{F}_2^{2k}.$$

- For $i \in [0, \ell - 1]$, $x_i = \text{Encode}(v'_i) \in \mathbb{F}_2^{2k}$.

Let $\mathbb{S} = (\mathbb{F}_2^k)^\ell \times (\mathbb{F}_2^k)^\ell \times \mathbb{F}_2^\ell$. For each

$$\phi = (b_1 \parallel \dots \parallel b_\ell \parallel e_1 \parallel \dots \parallel e_\ell \parallel g_1 \parallel \dots \parallel g_\ell) \in \mathbb{S},$$

it is possible to define the permutation Γ_ϕ that transforms

$$w^* = (y_1^* \parallel \dots \parallel y_\ell^* \parallel z_1^* \parallel \dots \parallel z_\ell^* \parallel x_1^* \parallel \dots \parallel x_{\ell-1}^*) \in \mathbb{F}_2^\delta$$

with $y_i^*, z_i^* \in \mathbb{F}_2^k$ for $i \in [0, \ell]$ and $x_i^* \in \mathbb{F}_2^{2n}$ for $i \in [0, \ell - 1]$ into

$$\Gamma_\phi(w^*) = (\psi_{g_1, b_1}(y_1^*) \parallel \dots \parallel \psi_{g_\ell, b_\ell}(y_\ell^*) \parallel \psi_{g_1, b_1}(z_1^*) \parallel \dots \parallel \psi_{g_\ell, b_\ell}(z_\ell^*) \parallel \psi_{g_1, b_1}(x_1^*) \parallel \dots \parallel \psi_{g_{\ell-1}, b_{\ell-1}}(x_{\ell-1}^*)).$$

At this point it is clear that the conditions in (7.1) are satisfied and the relation R_{acc} is reduced to $R_{abstract}$ (7.2).

7.5.2 An Update Algorithm

This update Algorithm is similar to the one presented in [97]. Given a Merkle tree cryptographic accumulator τ , an integer j with his binary representation $(j_1, \dots, j_\ell)$ and a value d' , TUpdate (Algorithm 7.9) allow us to update the accumulated values without reconstruct the whole tree. Indeed, to update $\mathcal{T}$ of $N = 2^\ell$ accumulated values we just have to update ℓ values as shown in Algorithm 7.9.

Algorithm 7.9 TUpdate() Algorithm

Require: A Merkle tree cryptographic accumulator $\mathcal{T}$, the binary representation $(j_1, \dots, j_\ell)$ of an integer j and a value d' .

Ensure: The updated Merkle tree $\mathcal{T}$.

Let d_j be the value at the position j in τ and $((j_1, \dots, j_\ell), (w_\ell, \dots, w_1))$ be its associated witness.

Set $v_\ell = d'$ and recursively compute the path $v_\ell, v_{\ell-1}, \dots, v_1, v_0 \in \mathbb{F}_2^k$ as in Algorithm 7.6.

Set $A_{\mathcal{T}} = v_0; y_{j_1} = v_1; \dots; y_{j_1, j_2, \dots, j_{\ell-1}} = v_{\ell-1}; y_{j_1, j_2, \dots, j_\ell} = v_\ell = d'$.

7.5.3 Description of Our *FDGS* Scheme

Let N and ℓ be integers where $N = 2^\ell - 1$ is the size of the group. The group manager GM chooses uniformly random vectors $x_0, \dots, x_{N-1} \in \mathbb{F}_2^n$ and respectively hashes them to $p_0, \dots, p_{N-1} \in \mathbb{F}_2^k$. Thereafter, GM uses our cryptographic accumulator to build a tree $\mathcal{T}$ on top with $(p_i)_{0 \leq i \leq N-1}$ and let $u \in \mathbb{F}_2^k$ be the accumulated value. At the beginning, all the $(x_i)_{0 \leq i \leq N-1}$ and $(p_i)_{0 \leq i \leq N-1}$ value are set to 0. It is when a user i joins or is active in the group that the associate values (x_i, p_i) are become a non-null. To generate a valid group signature, the user i have to prove the knowledge of a valid pair (x_i, p_i) and the witness of p_i . Also, using the Randomized QC-MDPC McEliece Encryption Scheme, the user i have to encrypt the binary representation of i and prove that the ciphertext is well-formed. We describe our protocol as follow:

GSetup(λ) The setup algorithm described in algorithm 7.10, take in input the security parameter λ and output the public parameter p_p .

Algorithm 7.10 GSetup(λ) Algorithm

Require: A security parameter λ .

Ensure: a public parameter p_p .

- The GM chooses tow polynomially bound positive integers N and ℓ such that $N = 2^\ell - 1$ where N is the capacity of our group.
 - Pick $(k_A, n_A, t_A) \in \mathbb{N}^3$, such that there exist an (n_A, k_A, t_A) double circulant code and Problems 6 and 7 get a security level of λ bits where $n_A = 2k_A$.
 - Pick $(n_M, q_M, w_M, t_M) \in \mathbb{N}^4$, such that the Randomized QC-MDPC McEliece Encryption Scheme describe in Section 3.2.2 get a security level of λ bits.
 - Pick a hash function $\mathcal{H}_{FS} : \{0, 1\}^* \rightarrow \{1, 2, 3\}^\sigma$ to be modelled as a random oracle in the Fiat-Shamir transformations. Here, σ is the number of round needed in the Fiat-Shamir transformation to get a security level ok λ bits.
 - Let COM be the string commitment scheme defined in Section 7.5.1, to be used in our zero-knowledge argument systems.
- return** $p_p = (k_A, n_A, t_A, n_M, q_M, w_M, t_M, N, \ell, \mathcal{H}_{FS}, \text{COM})$
-

$\langle \text{GKgen}_{\text{GM}}(p_p), \text{GKgen}_{\text{TM}}(p_p) \rangle$. In this interactive protocol, described in Algorithm 7.11, the group manager GM and the tracing manager TM initialize their keys and the public group information.

Algorithm 7.11 $\langle \text{GKgen}_{\text{GM}}(\mathbf{p}_p), \text{GKgen}_{\text{TM}}(\mathbf{p}_p) \rangle$ Algorithm**Require:** A public parameter $\mathbf{p}_p$.

- $\text{GKgen}_{\text{GM}}(\mathbf{p}_p)$. GM, by using k_A, n_A, t_A , run Gen algorithm, get the matrix H and set $\mathbf{m}_{\text{pk}} = H_A \in \mathbb{F}_2^{q_M \times n_M}$ and set $\mathbf{m}_{\text{sk}} = \emptyset$.
- $\text{GKgen}_{\text{TM}}(\mathbf{p}_p)$. TM uses n_M, q_M, w_M, t_M to run the setup algorithm of the Randomized QC-MDPC McEliece Encryption Scheme describe in Section ?? and get the parity-check matrix $H_M \in \mathbb{F}_2^{q_M \times n_M}$ and its corresponding generator matrix $G_M \in \mathbb{F}_2^{(n_M - q_M) \times n_M}$ in row reduced echelon form. Then, TM sets $\mathbf{t}_{\text{sk}} = H_M$ and $\mathbf{mdk} = G_M$
- TM send $\mathbf{t}_{\text{pk}}$ to GM who initialise as follow:
 - Table $\mathbf{r} = (\mathbf{r}[0][1], \mathbf{r}[0][2], \dots, \mathbf{r}[N-1][1], \mathbf{r}[N-1][2])$. For each $i \in [0, N-1]$: $\mathbf{r}[i][1] = 0^{k_A}$ will be used to store the registered user public key and $\mathbf{r}[i][2] = 0$ will be used to store the epoch at which the user joins.
 - The Merkel tree $\mathcal{T}$ built on top of $\mathbf{r}[0][1], \dots, \mathbf{r}[N-1][1]$.
 - Counter of registered users $c = 0$

Then, GM output $\mathbf{g}_{\text{pk}} = (\mathbf{p}_p, \mathbf{m}_{\text{pk}}, \mathbf{t}_{\text{pk}})$ and publishes the initial information $\mathcal{I} = \emptyset$. He keeps $\mathcal{T}$ and c as secret.

UKgen($\mathbf{p}_p$). In this algorithm (Algorithm 7.12), each potential group user can generate its key pair.

Algorithm 7.12 UKgen($\mathbf{p}_p$) Algorithm**Require:** A public parameter $\mathbf{p}_p$.**Ensure:** The user key pair $(\mathbf{u}_{\text{pk}}, \mathbf{u}_{\text{sk}})$.

- The user picks a non-zero random vector $x \in \mathbb{F}_2^{n_A}$ and computes $p = H_A \cdot \text{RE}(x)$.
- Set $\mathbf{u}_{\text{pk}} = p$ and $\mathbf{u}_{\text{sk}} = x$

GUpdate($\mathbf{g}_{\text{pk}}, \mathbf{m}_{\text{sk}}, \mathcal{I}, \mathbb{S}, \mathbf{r}$). This algorithm (Algorithm 7.13) is run by GM to update the group information while advancing the epoch.

Algorithm 7.13 $\text{GUpdate}(\mathbf{g}_{\text{pk}}, \mathbf{m}_{\text{sk}}, \mathcal{I}_\tau, \mathbb{S}, \mathbf{r})$ Algorithm

Require: the current group information $\mathcal{I}_\tau$, the group public key $\mathbf{g}_{\text{pk}}$, the GM secret key $\mathbf{m}_{\text{sk}}$, a table $\mathbb{S}$ containing the public keys of registered users to be revoked and the table $\mathbf{r}$.

Ensure: $\mathcal{I}_{\tau_{\text{new}}}$.

1. if $\mathbb{S} = \emptyset$ then go to Step 2.
Otherwise, $\mathbb{S} = \{\mathbf{r}[i_1][1], \dots, \mathbf{r}[i_r][1]\}$, for some $t \in [1, N]$ and some $i_1, \dots, i_r \in [0, N - 1]$. Then, for all $t \in [0, r]$, GM runs $\text{TUpdate}(\text{bin}(i_t), 0^{n_A})$ to update the tree $\mathcal{T}$.
2. For each active user $j \in [0, N - 1]$, let $w_j \in \mathbb{F}_2^\ell \times (\mathbb{F}_2^k)$ be the witness for the fact that p_j is accumulated in $u_{\tau_{\text{new}}}$. Then GM publishes the group information of the new epoch as:

$$\mathcal{I}_{\tau_{\text{new}}} = (u_{\tau_{\text{new}}}, \{w_j\}_j)$$

IsActive($\mathcal{I}_\tau, \mathbf{r}, u_{\text{id}}$). In this algorithm (Algorithm 7.14), if the user u_{id} is active at epoch τ we outputs 1 and 0 otherwise.

Algorithm 7.14 $\text{IsActive}(\mathcal{I}_\tau, \mathbf{r}, u_{\text{id}})$ Algorithm

Require: $\mathcal{I}_\tau, \mathbf{r}, u_{\text{id}}$.

Ensure: A value 0 or 1.

- if $\mathbf{r}[u_{\text{id}}][2] \neq \tau$ or $\mathbf{r}[u_{\text{id}}][1] = 0_A^k$ then return 0.
 - Else, return 1.
-

Join($\mathcal{I}_\tau, \mathbf{g}_{\text{pk}}, \mathbf{u}_{\text{pk}}, \mathbf{u}_{\text{sk}}$). In this algorithm defined in Algorithm 7.15, the user interact with the GM to join the group at epoch τ

Algorithm 7.15 $\text{Join}(\mathcal{I}_\tau, g_{pk}, u_{pk}, u_{sk})$ Algorithm

Require: the current group information $\mathcal{I}_\tau$, the group public key g_{pk} and the user key pair $(u_{pk}, u_{sk}) = (p, x)$

1. GM issues a member identifier for the user as $u_{id} = \text{bin}(c) \in \mathbb{F}_2^\ell$.
 2. The user sets his signing key as $g_{sk}[c] = (\text{bin}(c), p, x)$
 3. GM performs the following updates:
 - Updates the tree $\mathcal{T}$ by running $\text{TUpdate}(\text{bin}(c), p)$.
 - Register the user to table $\mathbf{r}$ as $\mathbf{r}[c][1] = p$ and $\mathbf{r}[c][2] = \tau$.
 - Set $c = c + 1$.
 - Set $\mathbb{S} = \emptyset$ and run GUpdate .
-

Issue $(\mathcal{I}_\tau, m_{sk}, u_{id})$. In Issue algorithm defined in Algorithm 7.16, the GM revokes the user with identifier u_{id} to the group at epoch τ .

Algorithm 7.16 $\text{Issue}(\mathcal{I}_\tau, m_{sk}, u_{id})$ Algorithm

Require: The current group information $\mathcal{I}_\tau$, the group manager secret key m_{sk} and the user identifier u_{id} .

1. GM sets $\mathbb{S} = \{\mathbf{r}[u_{id}][2]\}$.
 2. GM runs GUpdate .
-

Sign $(g_{pk}, g_{sk}(j), \mathcal{I}_\tau, M)$. Let a user with the tuple $g_{sk}(j) = (\text{bin}(j), p, x)$. To sign a message M using the group information at epoch τ , the user download u_τ and the witness $(\text{bin}(j), (w_\ell, \dots, w_1))$ from $\mathcal{I}_\tau$ and proceeds as in Algorithm 7.17.

Algorithm 7.17 $\text{Sign}(g_{pk}, gsk(j), \mathcal{I}_\tau, M)$ Algorithm

Require: $g_{pk}, gsk(j), \mathcal{I}_\tau, M$.

1. Run the function $\mathcal{E}(G_M, \text{bin}(j))$ defined in Section ?? to encrypt the vector $\text{bin}(j)$ and get the cipher text $s \in \mathbb{F}_2^{n_M}$.
2. Use the protocol defined in Section 7.5.1 to generate a Non-Interactive Zero-Knowledge Argument of Knowledge (NIZKAoK) Π_{gs} to demonstrate the possession of a valid tuple

$$\zeta = (x, p, \text{bin}(j), w_\ell, \dots, w_1) \quad (7.11)$$

such that $\text{TVerify}(H_A, u_\tau, p, (\text{bin}(j), (w_\ell, \dots, w_1))) = 1$, $H_A \cdot x = p$ and $p \neq 0^{k_A}$. Using the Fiat-Shamir heuristic, the protocol is repeat many σ times to achieve a negligible soundness error and get the triple: $\Pi_{gs} = (\{C\}_{i=1}^\sigma, e, \{R\}_{i=1}^\sigma)$, where $e = \mathcal{H}_{FS}(M, \{C\}_{i=1}^\sigma, u_\tau, s, H_A, G_M)$.

3. Output the group signature

$$\Sigma = (\Pi_{gs}, s). \quad (7.12)$$

Verify $(g_{pk}, \mathcal{I}_\tau, M, \Sigma)$. This algorithm (Algorithm 7.18) uses the global public key g_{pk} and the group information at epoch τ to verify the signature Σ of the message M .

Algorithm 7.18 $\text{Verify}(g_{pk}, \mathcal{I}_\tau, M, \Sigma)$ Algorithm

Require: $g_{pk}, \mathcal{I}_\tau, M, \Sigma$.

1. Get $u_\tau \in \mathbb{F}_2^{k_A}$ from $\mathcal{I}_\tau$.
 2. Parse Σ as $\Sigma = (\{C\}_{i=1}^\sigma, (e_1, \dots, e_\sigma), \{R\}_{i=1}^\sigma, s)$.
 3. If $(e_1, \dots, e_k) \neq \mathcal{H}_{FS}(M, \{C\}_{i=1}^\sigma, u_\tau, s, H_A, G_M)$, then return 0.
 4. For each $i = 1$ to σ , check the validity of R_i with respect to C_i and e_i by running the verification phase of the Stern-like protocol (Algorithm 7.2). If any of the condition does not hold, then return 0.
 5. Return 1.
-

Trace($\mathbf{g}_{\text{pk}}, \mathbf{t}_{\text{sk}}, \mathcal{J}_\tau, \mathbf{r}, M, \Sigma$). In this algorithm (Algorithm 7.19), TM uses his secret key $\mathbf{t}_{\text{sk}} = H_M$ to reveal the u_{id} of the signer.

Algorithm 7.19 $\text{Trace}(\mathbf{g}_{\text{pk}}, \mathbf{t}_{\text{sk}}, \mathcal{J}_\tau, \mathbf{r}, M, \Sigma)$ Algorithm

Require: $\mathbf{g}_{\text{pk}}, \mathbf{t}_{\text{sk}}, \mathcal{J}_\tau, \mathbf{r}, M, \Sigma$.

1. TM parses Σ as in (7.12) and runs the function $\mathcal{D}(H_M, s)$ defined in Section ?? to get $b' \in \mathbb{F}_2^\ell$.
 2. If $\mathcal{J}_\tau$, does not include a witness containing b' , then return *null*.
 3. Let $j' \in [0, N - 1]$ be the integer having binary representation b' . If the record $\mathbf{r}[j'][1]$ is 0^{k_A} , then return *null*.
 4. TM uses the Stern-like protocol defined in Section 7.2 to generate a Non-Interactive Zero-Knowledge Argument of Knowledge (NIZKAoK) Π_{trace} to demonstrate the knowledge of $H_M \in \mathbb{F}_2^{q_M \times n_M}$ and $y \in \mathbb{F}_2^{k_M}$, such that $y = H_M \cdot (b' \parallel 0^{n_M - \ell})$. Using the Fiat-Shamir heuristic, the protocol is repeated many σ times to achieve a negligible soundness error and get the triple: $\Pi_{\text{trace}} = (\{C\}_{i=1}^\sigma, e, \{R\}_{i=1}^\sigma)$, where $e = \mathcal{H}_{FS}(M, \{C\}_{i=1}^\sigma, \mathcal{J}_\tau, \mathbf{g}_{\text{pk}}, \Sigma, b')$.
 5. Set $u_{\text{id}} = b'$
 6. Output $(u_{\text{id}}, \Pi_{\text{trace}})$
-

Judge($\mathbf{g}_{\text{pk}}, u_{\text{id}}, \mathcal{J}_\tau, \Pi_{\text{trace}}, M, \Sigma$). In this algorithm (Algorithm 7.20), we verify the argument Π_{trace} .

Algorithm 7.20 Judge($g_{pk}, u_{id}, \mathcal{J}_\tau, \Pi_{trace}, M, \Sigma$) Algorithm

Require: $g_{pk}, u_{id}, \mathcal{J}_\tau, \Pi_{trace}, M, \Sigma$.

1. Get $u_\tau \in \mathbb{F}_2^{k_A}$ from $\mathcal{J}_\tau$.
 2. Parse Π_{trace} , as $\Pi_{trace} = (\{C\}_{i=1}^\sigma, e, \{R\}_{i=1}^\sigma)$.
 3. If $(e_1, \dots, e_\sigma) \neq \mathcal{H}_{FS}(M, \{C\}_{i=1}^\sigma, \mathcal{J}_\tau, g_{pk}, \Sigma, b')$, then return 0.
 4. For each $i = 1$ to σ , check the validity of R_i with respect to C_i and e_i by running the verification phase of the Stern-like protocol (Algorithm 7.2). If any of the condition does not hold, then return 0.
 5. Return 1.
-

7.5.4 Analysis of Our *FDGS* Scheme

In this part we study the security requirements as it was defined in Algorithms 2.5, 2.6, 2.7, 2.8 and 2.9 of our *FDGS*. We start with correctness properties and finish with tracing soundness properties.

Correctness. We study the correctness of our *FDGS* in Theorem 12.

Theorem 12. *If there exist a PPT adversary $\mathcal{A}$ against the correctness property of our *FDGS* then we always have $Pr[Exp_{FDGS, \mathcal{A}}^{corr}(\lambda) = 1]$ close to 1.*

Proof. Let j be an active and honest user which have generate a signature $\Sigma = (\Pi_{gs}, s)$. This user j can always compute a tuple $\zeta = (x, p, \text{bin}(j), w_\ell, \dots, w_1)$ satisfying the condition in Sign algorithm. Then, the completeness of the underlying argument system grants that Σ is always accepted by the Verify algorithm. Also, the description algorithm of the Randomized QC-MDPC McEliece Encryption Scheme, with a tiny rate failure, will output $\text{bin}(j)$ by decrypt s . Further, since the user j is active, $\mathcal{J}_\tau$ contain $w = (\text{bin}(j), w_\ell, \dots, w_1)$ and $r[j][1] \neq 0$. Therefore, the Trace Algorithm can always obtain the tuple (H_M, y) satisfying the condition in the Trace Algorithm. By the completeness of the argument system, Π_{trace} will be accepted by the Judge algorithm. $\square$

Anonymity. We study the anonymity of our *FDGS* scheme in Theorem 13.

Theorem 13. *The FDGS scheme proposed in Section 7.5.3, provide the anonymity in the random oracle model assuming the hardness of DCRSD problem (Problem 6).*

Proof. Let $\mathcal{A}$ and $\mathcal{B}$ be a PPT algorithms that act respectively as adversary and challenger. We construct a sequence of indistinguishable experiments beginning with $Exp_{FDGS, \mathcal{A}}^{anon-0}(\lambda)$ to finish with $Exp_{FDGS, \mathcal{A}}^{anon-1}(\lambda)$. For each experiment Exp_i , we call W_i the output of the adversary.

Exp₀: This experiment is the same as the experiment $Exp_{FDGS, \mathcal{A}}^{anon-0}(\lambda)$.

Exp₁ This experiment is the same as in Exp_0 except that it simulates a proof for Trace oracle by programming the random oracle $\mathcal{H}_{FS}$ although the challenger $\mathcal{B}$ has the correct witness to generate a valid proof. At this point, given the statistical zero-knowledge property of our argument system generating Π_{trace} , the view of the adversary $\mathcal{A}$ is statistically indistinguishable between Exp_0 and Exp_1 . Then it is clear the $Pr[W_0 = 1]$ is close to $Pr[W_1 = 1]$.

Exp₂ This experiment generate a simulated proof for $Chal_b$ oracle. Given the statistical zero-knowledge property of our argument system generating Π_{gs} , the view is indistinguishable to the adversary $\mathcal{A}$. Then we have that $Pr[W_1 = 1]$ is close to $Pr[W_2 = 1]$.

Exp₃ This experiment is the same as Exp_2 except that s is now the encryption of $\text{bin}(i_1)$ for the $Chal_b$ query. Given the semantic security of Randomized QC-MDPC McEliece Encryption Scheme, this change is negligible for the adversary $\mathcal{A}$. Then we have

$$|Pr[W_2 = 1] - Pr[W_3 = 1]|$$

is negligible.

Exp₄ We now generate a real proof for the $Chal_b$ query instead of simulated proof. Here, the view of the adversary $\mathcal{A}$ in Exp_3 and Exp_4 are statistically indistinguishable because of the statistical zero-knowledge property of our argument system generating Π_{gs} . Then we have that $Pr[W_3 = 1]$ is close to $Pr[W_4 = 1]$.

Exp₅ In this Experiment, we generate a real proof for the Trace query. Here, the view of the adversary $\mathcal{A}$ in Exp_4 and Exp_5 are statistically indistinguishable because of the statistical zero-knowledge property of our argument system generating Π_{trace} . Then we have

$Pr[W_4 = 1]$ is close to $Pr[W_5 = 1]$. This experiment is the experiment $Exp_{FDGS, \mathcal{A}}^{anon-1}(\lambda)$. Hence we have $Pr[W_5 = 1] = Pr[Exp_{FDGS, \mathcal{A}}^{anon-1}(\lambda) = 1]$.

Given all the experiment above, we have that:

$$\left| Pr[Exp_{FDGS, \mathcal{A}}^{anon-1}(\lambda) = 1] - Pr[Exp_{FDGS, \mathcal{A}}^{anon-0}(\lambda) = 1] \right|$$

is negligible, hence our scheme provide the anonymity. $\square$

Non-Frameability. We study the non-frameability of our *FDGS* scheme in Theorem 14.

Theorem 14. *Our FDGS scheme provides the non-frameability property in the random oracle model since the 2-DCRNSD problem (Problem 7) is hard.*

Proof. Let $\mathcal{A}$ and $\mathcal{B}$ be respectively an adversary against the non-frameability of our *FDGS* scheme and a *PPT* algorithm that try to breaks the security of our cryptographic accumulator or solve 2-DCRNSD problem (Problem 7). We say that $\mathcal{A}$ succeed in the experiment $Exp_{FDGS, \mathcal{A}}^{non-frame}(\lambda)$ if the output of $Exp_{FDGS, \mathcal{A}}^{non-frame}(\lambda)$ is 1 and we define the advantage of $\mathcal{A}$ in $Exp_{FDGS, \mathcal{A}}^{non-frame}(\lambda)$ the probability $Pr[Exp_{FDGS, \mathcal{A}}^{non-frame}(\lambda) = 1]$. In the following we assume that $\mathcal{A}$ succeed in $Exp_{FDGS, \mathcal{A}}^{non-frame}(\lambda)$ with a non-negligible advantage ε . By the execution of the $Exp_{FDGS, \mathcal{A}}^{non-frame}(\lambda)$, the adversary $\mathcal{A}$ outputs the tuple $(M^*, \Sigma^*, \text{bin}(j^*), \Pi_{trace}^*, \mathcal{I}_\tau)$ then $\mathcal{B}$ proceed as follow:

Parse Σ^* as (Π_{gs}^*, s^*) , where $\Pi_{gs}^* = (\{C_i^*\}_{i=1}^\sigma, e^*, \{R_i^*\}_{i=1}^\sigma)$ and R_i^* is a valid response according to C_i^* and e_i^* for $i \in [1, \sigma]$ since $\mathcal{A}$ succeed. It is clear that $\mathcal{A}$ get the tuple $\zeta^* = (M^*, \{C_i^*\}_{i=1}^\sigma, u_\tau, s^*, H_A, G_M)$ from the hash function $\mathcal{H}_{FS}$ with overwhelming probability. Otherwise, $\mathcal{A}$ cans guess this value with a negligible probability of $3^{-\sigma}$. Therefore, the tuple ζ^* has been an input of one hash query with a probability of $\varepsilon' = \varepsilon - 3^{-\sigma}$. Let $t^* \in \{1, 2, \dots, q_{\mathcal{H}}\}$ be an integer, where $q_{\mathcal{H}}$ is the total number of hash queries made by $\mathcal{A}$ and m^* be a polynomially bounded integer in t^* . By the Forking Lemma (Theorem 9), $\mathcal{B}$ outputs ζ^* at the m^* -th query as in the original run. And with a probability $\geq 1/2$, $\mathcal{B}$ obtains 3-fork involving the same tuple ζ^* with pairwise distinct hash value $e_{m^*}^{(1)}, e_{m^*}^{(2)}, e_{m^*}^{(3)} \in \{1, 2, 3\}^\sigma$. At the j -th round of the execution of the argument of knowledge the system, for some $j \in \{1, 2, \dots, \sigma\}$, we have $(e_{m^*, j}^{(1)}, e_{m^*, j}^{(2)}, e_{m^*, j}^{(3)}) = (1, 2, 3)$ with the probability of $1 - (\frac{1}{9})^\sigma$. Then by the argument of knowledge of the system generating Π_{gs} , from the valid responses $(e_{m^*, j}^{(1)}, e_{m^*, j}^{(2)}, e_{m^*, j}^{(3)})$, we can extract the witness $\zeta' = (x', p', w'_\tau)$ where

$$w'_\tau = (\text{bin}(j'), w'_{\ell, \tau}, \dots, w'_{1, \tau}) \in \mathbb{F}_2^\ell \times (\mathbb{F}_2^{k_A})$$

such that:

1. $\text{TVerify}(H_A, u_\tau, p', w'_\tau) = 1$
2. $p' = H_A \cdot \text{RE}(x')$
3. s^* is the encryption of $\text{bin}(j')$.

By the correctness of our encryption scheme, s^* is the decrypted to $\text{bin}(j')$. The fact that $\mathcal{A}$ succeed in the experiment $\text{Exp}_{FDGS, \mathcal{A}}^{\text{non-frame}}(\lambda)$ imply that the Judge algorithm output 1. By the soundness of our argument of knowledge of the system generating Π_{gs} , $\text{bin}(j^*)$ is decrypted from s^* . Then, we have $\text{bin}(j') = \text{bin}(j^*)$. We consider now the following cases:

1. $\text{bin}(j')$ is an inactive user at epoch τ , it means the value at leaf $\text{bin}(j')$ is null. Which is not valid since we have $\text{TVerify}(H_A, u_\tau, p', w'_\tau) = 1$ with $p' \neq 0$ at leaf $\text{bin}(j')$.
2. $\text{bin}(j')$ is an active user at epoch τ . The fact that $\text{bin}(j^*) \in \text{HUL} \setminus \text{BUL}$ means $\mathcal{A}$ does not know $g_{sk}[j^*] = (\text{bin}(j^*), p', x_{j^*})$, where x_{j^*} was initially chosen by $\mathcal{B}$ such that $p' = H_A \cdot \text{RE}(x_{j^*})$. With a probability $\geq \frac{1}{2}$ we have $x_{j^*} \neq x'$. Then we solve the 2-DCRNSD problem (Problem 7) with a non-negligible probability if $\mathcal{A}$ breaks the non-frameability of our construction with non-negligible probability.

□

Traceability. We study the traceability of our *FDGS* scheme in Theorem 15.

Theorem 15. *Our FDGS scheme provides the traceability property in the random oracle model since the 2-DCRNSD problem (Problem 7) is hard.*

Proof. We say that an adversary $\mathcal{A}$ succeed in traceability experiment described in $\text{Exp}_{FDGS, \mathcal{A}}^{\text{trace}}(\lambda)$ if it generates a valid signature that either:

Case 1: traces to an inactive user;

Case 2: traces to an active user but we cannot generate a proof accepted by the Judge algorithm.

In the following, we prove that both cases may occur with negligible probability. Let $(\mathcal{I}_\tau, M, \Sigma)$ be the output of $\mathcal{A}$ in $Exp_{FDGS, \mathcal{A}}^{trace}(\lambda)$ and by running the algorithm Trace we output $(\text{bin}(j), \Pi_{trace})$. After, we parse Σ as (Π_{gs}, s) where $\Pi_{gs} = (\{C_i\}_{i=1}^\sigma, e, \{R_i\}_{i=1}^\sigma)$ and R_i^* is a valid response according to C_i and e_i for $i \in [1, \sigma]$ since $(\mathcal{I}_\tau, M, \Sigma)$ is a valid signature generated by $\mathcal{A}$. It is the same situation as in Theorem 14. Then, using the same technique we extract the witness $\zeta = (x, p, w_\tau)$ where

$$w_\tau = (\text{bin}(j'), w_{\ell, \tau}, \dots, w_{1, \tau}) \in \mathbb{F}_2^\ell \times (\mathbb{F}_2^{k_A})$$

such that:

1. $\text{TVerify}(H_A, u_\tau, p, w_\tau) = 1$
2. $p \neq 0$
3. s is the encryption of $\text{bin}(j')$.

By the correctness of our encryption scheme, s is the decrypted to $\text{bin}(j')$. The correct decryption of the Trace algorithm, which is run by the challenger, indicate that $\text{bin}(j)$ is decrypted from s . Hence we have $\text{bin}(j') = \text{bin}(j)$.

We stress that **Case 1**, defined above, will occur with negligible probability. Because, if $\text{bin}(j)$ is not active at epoch τ , and $\text{TVerify}(H_A, u_\tau, p, w_\tau)$ with $p \neq 0$ at leaf $\text{bin}(j)$, then the security of our cryptographic accumulator is violated. Furthermore, **Case 2**, defined above, will occur with negligible probability since the challenger possesses the valid witness to generate the proof Π_{trace} , which will be accepted by the Judge algorithm, thank to the completeness to the underlying zero-knowledge argument system. $\square$

Tracing Soundness. We study the tracing soundness of our *FDGS* scheme in Theorem 16.

Theorem 16. *Our FDGS scheme provides the tracing soundness property in the random oracle model.*

Proof. Let $(M, \Sigma, \text{bin}(j_0), \Pi_{trace,0}, \text{bin}(j_1), \Pi_{trace,1}, \mathcal{I}_\tau)$ be the output of the adversary $\mathcal{A}$ in the experiment $Exp_{FDGS, \mathcal{A}}^{trace-sound}(\lambda)$. We say that an adversary $\mathcal{A}$ succeed in tracing soundness experiment described in $Exp_{FDGS, \mathcal{A}}^{trace}(\lambda)$ if the following conditions is hold:

1. $\text{bin}(j_0) \neq \text{bin}(j_1)$;

2. $\text{Verify}(\mathbf{g}_{\text{pk}}, \mathcal{T}_\tau, M, \Sigma) = 1$;
3. $\text{Judge}(\mathbf{g}_{\text{pk}}, \text{bin}(j_b), \mathcal{T}_\tau, \Pi_{\text{trace}, b}, M, \Sigma) = 1$ for $b \in \{0, 1\}$;
4. $\text{bin}(j_b) \neq \text{null}$ for $b \in \{0, 1\}$.

We parse $\Pi_{\text{trace}, b}$ as $(\{C_{i,b}\}_{i=1}^\sigma, e_b, \{R_{i,b}\}_{i=1}^\sigma)$. We note that $R_{i,b}$ is a valid response according to $C_{i,b}$ and $e_{i,b}$ for $i \in [1, \sigma]$ and $b \in \{0, 1\}$ since the Judge algorithm output 1. Thereafter, it means that s is both the encryption of $\text{bin}(j_0)$ and $\text{bin}(j_1)$ which is absurd. Then our construction satisfies the tracing soundness property. $\square$

7.5.5 Efficiency

In this section, we analyse the efficiency of our *FDGS* scheme and we propose a parameters to instantiate our scheme for a group of size $N = 2^\ell - 1$.

- The size of the group public key $\mathbf{g}_{\text{pk}}$ is about $k_A + q_M$ bits, where k_A and q_M are as in Algorithm 7.10.
- For each signer $j \in [0, N - 1]$, the size of the signing key is about $\ell + 3k_A$ bits
- At each epoch, the signature verifiers download k_A bits, while each active signer downloads $\ell(1 + k_A)$ bits.
- The size of a signature Σ is about $\sigma(7\delta + 1) + n_M$ bits, where $\delta = 2k_A(3\ell - 1)$.
- The size of the Stern-Like Non-Interactive Zero-Knowledge Argument of Knowledge (NIZKAoK) Π_{trace} is about $\ell_{\mathcal{H}_{FS}} + \sigma\left(6k_A + \frac{24k_A}{3}\right)$ bits. Where $\ell_{\mathcal{H}_{FS}}$ is the output size of $\mathcal{H}_{FS}$ function.

Its clear that the size of our group signature is logarithmic on the size of the group.

7.6 Conclusions

In this chapter, we have proposed a cryptographic accumulator based on coding theory assumptions which are resistant to the quantum computer. We have instantiated our scheme with double circulant codes, which provide for an 80 bits security a small public key of 347 bits. We have also proved the feasibility of our scheme by providing a first direct implementation of post-quantum cryptographic accumulators. Thereafter, we use our cryptographic accumulator to propose a code based fully dynamic group signature which is logarithmic on the size of the group. In future work, we will try to add the non-membership proof, to prove that an element is not accumulated in our accumulated value. We will also try to add dynamic properties to our cryptographic accumulator.

Conclusion and Perspectives

Conclusion

The combination of Shor's algorithm and the promised quantum computer have enabled the emergence of post-quantum cryptography. In this mind, the NIST launched a contest in 2016 to standardise one or more post-quantum cryptographic algorithms. Actually, in 2020, this contest is on its third round. In this thesis we bring our contribution in this research area and particularly in code-based cryptography. Our contribution consist in three topics : the first one is for group signatures the second one is for identification and authentication schemes and the third one is for cryptographic accumulators.

Groups signature was treated in Chapters 3 and 4. In Chapter 3, due to the main issue in code-based cryptography that is the keys size, we focus on the reduction of the public key size of the first provably secure code based group signature scheme[58]. We have proposed a provably secure code based group signature scheme with reduced public key length uses Quasi-cyclic Moderate Density Parity-Check (QC-MDPC) codes and satisfies the correctness, CPA anonymity and traceability properties which are the security requirements for a static group signature scheme. Du to its small key size, our proposed can be used in system with small memory like smart cards. In Chapter 4, by always using Quasi-cyclic Moderate Density Parity-Check (QC-MDPC) codes, we have proposed a new dynamic group signature scheme based on coding theory assumptions which improve the first code based group signature presented in [3]. In this proposed scheme, group public key and signatures sizes remain independent of group length but just depending on the security level. Our technique consist in using the QC-MDPC version of McEliece cryptosystem in the join protocol and AGS identification scheme as Zero-Knowledge argument system unlike using the trapdoor matrix and Stern identification scheme as described in [3]. These choices make the join protocol very fast and decreases the public key and signature length. However, our proposed like the first code based group

signature presented in [3] is not fully dynamic since it is not possible to delete a members form the member group.

Identification and authentication schemes was treated in Chapter 5 and 6. In Chapter 5, we have proposed an efficient Zero-knowledge identification scheme with a cheating probability of $\frac{1}{2}$ based on coding theory assumptions which is supposed to be resistant to quantum computers. We have used rank metric rather than the classical Hamming metric. This choice implies a significant reduction in public data size and in communication cost. It is also possible to turn it into a signature scheme by applying the Fiat-Schamir Transformation described in [62]. Du to its small key size and communication cost, our proposed can be used in system with small memory like smart card. And In Chapter 6, we have proposed a mutual authentication protocol based on hard problems from double circulant low rank parity check codes. These problems are supposed to be resistant to quantum computing attacks. Our proposed protocol meets all important security and privacy requirements. In addition to classic security and privacy requirements for RFID systems, our scheme achieves the full inside secrecy property. Also, our protocol saves at least 42% in terms of communication data compared to the most recent code-based authentication protocols for RFID systems.

To finish, cryptographic accumulators was treated in Chapter 7, here, we have proposed a cryptographic accumulator based on coding theory assumptions which are resistant to the quantum computer. We have instantiated our scheme with double circulant codes, which provide for an 80 bits security a small public key of 347 bits. We have also proved that the feasibility of our scheme by providing a first direct implementation of post-quantum cryptographic accumulators. However, this first code-based cryptographic accumulator is not dynamic. Thereafter, we use our cryptographic accumulator to proposed a code based fully dynamic group signature which is logarithmic on the size of the group.

Perspectives

The big issue in code-based cryptography remains the size of data. In this thesis, we have focus on the reducing of the key and the signature sizes of code-based group signature but it remaining a lot of progress to do. Clearly, the rank metric turn out to be an interesting choice on the managed of amount of data needed. Thus, it would be interesting in future works to focus on construction of rich cryptographic applications like group signature, ring signature using rank metric double circulant codes and also provide an implementation on smart card of our proposed schemes. It would require a big works on the security of using and metric double

circulant codes since the Rank Syndrome Decoding problem have not yet proved NP-complete. Thereafter, the used of cryptographic accumulators can be very important for saved memory when we have to manage a big set of values in constraint environment with small memory like an RFID Tags. So it would be interesting to focus also on designing of authentication for RFID Tags that uses code-based cryptographic accumulators. Also, using our proposed code-based cryptographic accumulators we will focus on designing fully dynamic code-based group signatures, code-based redactable signatures, code-based sanitizable signatures and code-based identity-based encryption.

References

- [1]C. M. Aguilar, P. Gaborit, and J. Schrek. “A new zero-knowledge code based identification scheme with reduced communication”. In: *ITW 2011, Paraty, Brazil, October 16-20*. IEEE, 2011, pp. 648–652. URL: <https://doi.org/10.1109/ITW.2011.6089577>.
- [2]M. Ajtai. “Generating Hard Instances of Lattice Problems (Extended Abstract)”. In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*. ACM, 1996, pp. 99–108. URL: <https://doi.org/10.1145/237814.237838>.
- [3]Q. Alam  lou, O. Blazy, S. Cauchie, and P. Gaborit. “A code-based group signature scheme”. In: *Des. Codes Cryptogr.* 82.1-2 (2017), pp. 469–493. URL: <https://doi.org/10.1007/s10623-016-0276-6>.
- [4]S. M. E. Y. Alaoui,  . Dagdelen, P. V  ron, D. Galindo, and P. Cayrel. “Extended Security Arguments for Signature Schemes”. In: *AFRICACRYPT 2012, Ifrance, Morocco, July 10-12*. Vol. 7374. Lecture Notes in Computer Science. Springer, 2012, pp. 19–34. URL: https://doi.org/10.1007/978-3-642-31410-0_2.
- [5]N. Aragon, P. Gaborit, A. Hauteville, O. Ruatta, and G. Z  mor. “Low Rank Parity Check Codes: New Decoding Algorithms and Applications to Cryptography”. In: *IEEE Trans. Inf. Theory* 65.12 (2019), pp. 7697–7717. URL: <https://doi.org/10.1109/TIT.2019.2933535>.
- [6]N. Aragon, P. Gaborit, A. Hauteville, and J. Tillich. “A New Algorithm for Solving the Rank Syndrome Decoding Problem”. In: *2018 IEEE International Symposium on Information Theory, ISIT 2018, Vail, CO, USA, June 17-22, 2018*. IEEE, 2018, pp. 2421–2425. URL: <https://doi.org/10.1109/ISIT.2018.8437464>.
- [7]N. Aragon, P. Gaborit, A. Hauteville, and J. Tillich. “A New Algorithm for Solving the Rank Syndrome Decoding Problem”. In: *2018 IEEE International Symposium on Information Theory, ISIT 2018, Vail, CO, USA, June 17-22, 2018*. IEEE, 2018, pp. 2421–2425. URL: <https://doi.org/10.1109/ISIT.2018.8437464>.

References

- [8]H. Assidi, E. B. Ayebie, and E. M. Souidi. “A Code-based Group Signature Scheme with Shorter Public Key Length”. In: *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications (ICETE 2016) - Volume 4: SECRIPT, Lisbon, Portugal, July 26-28, 2016*. SciTePress, 2016, pp. 432–439. URL: <https://doi.org/10.5220/0005969204320439>.
- [9]H. Assidi, E. B. Ayebie, and E. M. Souidi. “A Code-based Group Signature Scheme with Shorter Public Key Length”. In: *SECRIPT 2016 - Lisbon, Portugal*. SciTePress, 2016, pp. 432–439. URL: <https://doi.org/10.5220/0005969204320439>.
- [10]H. Assidi, E. B. Ayebie, and E. M. Souidi. “Two Mutual Authentication Protocols Based on Zero-Knowledge Proofs for RFID Systems”. In: *ICISC 2017, Seoul, South Korea*. Vol. 10779. Lecture Notes in Computer Science. Springer, 2017, pp. 267–283. URL: https://doi.org/10.1007/978-3-319-78556-1_15.
- [11]G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. “A Practical and Provably Secure Coalition-Resistant Group Signature Scheme”. In: *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24*. Vol. 1880. Lecture Notes in Computer Science. Springer, 2000, pp. 255–270. URL: https://doi.org/10.1007/3-540-44598-6_16.
- [12]D. Augot, M. Finiasz, and N. Sendrier. “A Family of Fast Syndrome Based Cryptographic Hash Functions”. In: *Mycrypt 2005, Kuala Lumpur, Malaysia*. Vol. 3715. Lecture Notes in Computer Science. Springer, 2005, pp. 64–83. URL: https://doi.org/10.1007/11554868_6.
- [13]D. Augot, M. Finiasz, and N. Sendrier. “A Fast Provably Secure Cryptographic Hash Function”. In: *IACR Cryptology ePrint Archive (2003)*. URL: <http://eprint.iacr.org/2003/230>.
- [14]B. E. Ayebie, H. Assidi, and E. M. Souidi. “A New Dynamic Code-Based Group Signature Scheme”. In: *C2SI 2017, Rabat, Morocco*. Vol. 10194. Lecture Notes in Computer Science. Springer, 2017, pp. 346–364. URL: https://doi.org/10.1007/978-3-319-55589-8_23.
- [15]E. B. Ayebie, H. Assidi, and E. M. Souidi. “An Efficient Identification Scheme Based on Rank Metric”. In: *Foundations and Practice of Security - 12th International Symposium, FPS 2019, Toulouse, France, November 5-7, 2019, Revised Selected Papers*. Vol. 12056. Lecture Notes in Computer Science. Springer, 2019, pp. 273–289. URL: https://doi.org/10.1007/978-3-030-45371-8_17.
- [16]E. B. Ayebie and E. M. Souidi. “Rank-metric code-based mutual authentication protocol for RFID”. In: *Journal of Information Security and Applications* 55 (2020), p. 102598. URL: <http://www.sciencedirect.com/science/article/pii/S2214212620307638>.
- [17]K. Bagadia, U. Chatterjee, D. B. Roy, D. Mukhopadhyay, and R. S. Chakraborty. “Revisiting the Security of LPN Based RFID Authentication Protocol and Potential Exploits in Hardware Implementations”. In: *SPACE 2019, Gandhinagar, India*. Vol. 11947. Lecture Notes in Computer Science. Springer, 2019, pp. 214–230. URL: https://doi.org/10.1007/978-3-030-35869-3_15.

- [18]M. Bardet, P. Briaud, M. Bros, P. Gaborit, V. Neiger, O. Ruatta, and J. Tillich. “An Algebraic Attack on Rank Metric Code-Based Cryptosystems”. In: *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part III*. Vol. 12107. Lecture Notes in Computer Science. Springer, 2020, pp. 64–93. URL: https://doi.org/10.1007/978-3-030-45727-3_3.
- [19]M. Bardet, M. Bros, D. Cabarcas, P. Gaborit, R. A. Perlner, D. Smith-Tone, J. Tillich, and J. A. Verbel. “Algebraic attacks for solving the Rank Decoding and MinRank problems without Gröbner basis”. In: *CoRR abs/2002.08322* (2020). URL: <https://arxiv.org/abs/2002.08322>.
- [20]M. Bellare, D. Micciancio, and B. Warinschi. “Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions”. In: *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8*. Vol. 2656. Lecture Notes in Computer Science. Springer, 2003, pp. 614–629. URL: https://doi.org/10.1007/3-540-39200-9_38.
- [21]M. Bellare and P. Rogaway. “Random Oracles are Practical: A Paradigm for Designing Efficient Protocols”. In: *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993*. ACM, 1993, pp. 62–73. URL: <https://doi.org/10.1145/168588.168596>.
- [22]M. Bellare, H. Shi, and C. Zhang. “Foundations of Group Signatures: The Case of Dynamic Groups”. In: *Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18*. Vol. 3376. Lecture Notes in Computer Science. Springer, 2005, pp. 136–153. URL: https://doi.org/10.1007/978-3-540-30574-3_11.
- [23]E. Bellini, F. Caullery, P. Gaborit, M. Manzano, and V. Mateu. “Improved Veron Identification and Signature Schemes in the Rank Metric”. In: *IEEE, ISIT 2019, Paris, France, July 7-12*. IEEE, 2019, pp. 1872–1876. URL: <https://doi.org/10.1109/ISIT.2019.8849585>.
- [24]J. Benaloh and M. de Mare. “One-Way Accumulators: A Decentralized Alternative to Digital Signatures”. In: *Advances in Cryptology — EUROCRYPT '93*. Ed. by T. Helleseht. Vol. 765. Lecture Notes in Computer Science. Springer, 1994, pp. 274–285. URL: https://doi.org/10.1007/3-540-48285-7_24.
- [25]T. P. Berger. “Isometries for rank distance and permutation group of Gabidulin codes”. In: *IEEE Trans. Inf. Theory* 49.11 (2003), pp. 3016–3019. URL: <https://doi.org/10.1109/TIT.2003.819322>.
- [26]E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg. “On the inherent intractability of certain coding problems (Corresp.)” In: *IEEE Trans. Inf. Theory* 24.3 (1978), pp. 384–386. URL: <https://doi.org/10.1109/TIT.1978.1055873>.

References

- [27]D. J. Bernstein. “Grover vs. McEliece”. In: *Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28*. Vol. 6061. Lecture Notes in Computer Science. Springer, 2010, pp. 73–80. URL: https://doi.org/10.1007/978-3-642-12929-2_6.
- [28]D. J. Bernstein, T. Lange, and C. Peters. “Attacking and Defending the McEliece Cryptosystem”. In: *PQCrypto 2008, Cincinnati, OH, USA, October 17-19*. Vol. 5299. Lecture Notes in Computer Science. Springer, 2008, pp. 31–46. URL: https://doi.org/10.1007/978-3-540-88403-3_3.
- [29]O. Blazy. “Preuves de connaissances interactives et non-interactives. ((Non-)Interactive Proofs of Knowledge)”. PhD thesis. Paris Diderot University, France, 2012. URL: <https://tel.archives-ouvertes.fr/tel-00768787>.
- [30]D. Boneh, X. Boyen, and H. Shacham. “Short Group Signatures”. In: *Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*. Vol. 3152. Lecture Notes in Computer Science. Springer, 2004, pp. 41–55. URL: https://doi.org/10.1007/978-3-540-28628-8_3.
- [31]J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, and J. Groth. “Foundations of Fully Dynamic Group Signatures”. In: *14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016*. Vol. 9696. Lecture Notes in Computer Science. Springer, 2016, pp. 117–136. URL: https://doi.org/10.1007/978-3-319-39555-5_7.
- [32]X. Boyen and B. Waters. “Compact Group Signatures Without Random Oracles”. In: *EUROCRYPT 2006, St. Petersburg, Russia, May 28 - June 1*. Vol. 4004. Lecture Notes in Computer Science. Springer, 2006, pp. 427–444. URL: https://doi.org/10.1007/11761679_26.
- [33]P. Camacho, A. Hevia, M. Kiwi, and R. Opazo. “Strong accumulators from collision-resistant hashing”. In: *International Journal of Information Security* 11.5 (2012), pp. 349–363. URL: <https://dl.acm.org/doi/10.1007/s10207-012-0169-2>.
- [34]J. Camenisch, M. Kohlweiss, and C. Soriente. “An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials”. In: *PKC 2009, Irvine, CA, USA*. Vol. 5443. Lecture Notes in Computer Science. Springer, 2009, pp. 481–500. URL: https://doi.org/10.1007/978-3-642-00468-1_27.
- [35]J. Camenisch and A. Lysyanskaya. “Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials”. In: *CRYPTO 2002, Santa Barbara, California, USA*. Vol. 2442. Lecture Notes in Computer Science. Springer, 2002, pp. 61–76. URL: https://doi.org/10.1007/3-540-45708-9_5.
- [36]J. Camenisch and A. Lysyanskaya. “Signature Schemes and Anonymous Credentials from Bilinear Maps”. In: *Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19*. Vol. 3152. Lecture Notes in Computer Science. Springer, 2004, pp. 56–72. URL: https://doi.org/10.1007/978-3-540-28628-8_4.

- [37]J. Camenisch, G. Neven, and M. Rückert. “Fully Anonymous Attribute Tokens from Lattices”. In: *SCN 2012, Amalfi, Italy, September 5-7*. Vol. 7485. Lecture Notes in Computer Science. Springer, 2012, pp. 57–75. URL: https://doi.org/10.1007/978-3-642-32928-9_4.
- [38]J. Camenisch and M. Stadler. In: *CRYPTO 97, California, USA August 17–21*. Vol. 1294. Lecture Notes in Computer Science. Springer, 1997, pp. 410–424. URL: <https://doi.org/10.1007/BFb0052252>.
- [39]D. Catalano and D. Fiore. “Vector Commitments and Their Applications”. In: *PKC 2013, Nara, Japan*. Vol. 7778. Lecture Notes in Computer Science. Springer, 2013, pp. 55–72. URL: https://doi.org/10.1007/978-3-642-36362-7_5.
- [40]P. Cayrel, P. Véron, and S. M. E. Y. Alaoui. “A Zero-Knowledge Identification Scheme Based on the q-ary Syndrome Decoding Problem”. In: *Selected Areas in Cryptography - 17th International Workshop, SAC 2010, Waterloo, Ontario, Canada, August 12-13, 2010, Revised Selected Papers*. Vol. 6544. Lecture Notes in Computer Science. Springer, 2010, pp. 171–186. URL: https://doi.org/10.1007/978-3-642-19574-7_12.
- [41]F. Chabaud and J. Stern. “The Cryptographic Security of the Syndrome Decoding Problem for Rank Distance Codes”. In: *ASIACRYPT '96, Kyongju, Korea, November 3-7*. Vol. 1163. Lecture Notes in Computer Science. Springer, 1996, pp. 368–381. URL: <https://doi.org/10.1007/BFb0034862>.
- [42]D. Chaum and E. van Heyst. “Group Signatures”. In: *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*. Vol. 547. Lecture Notes in Computer Science. Springer, 1991, pp. 257–265. URL: https://doi.org/10.1007/3-540-46416-6_22.
- [43]C. L. Chen, Y. L. Lai, C. C. Chen, Y. Y. Deng, and Y. C. Hwang. “RFID Ownership Transfer Authorization Systems Conforming EPC global Class-1 Generation-2 Standards”. In: *I. J. Network Security* 13.1 (2011), pp. 41–48. URL: <http://ijns.femto.com.tw/contents/ijns-v13-n1/ijns-2011-v13-n1-p41-48.pdf>.
- [44]K. Chen. “A New Identification Algorithm”. In: *Cryptography: Policy and Algorithms, International Conference, Brisbane, Queensland, Australia, July 3-5*. Vol. 1029. Lecture Notes in Computer Science. Springer, 1995, pp. 244–249. URL: <https://doi.org/10.1007/BFb0032363>.
- [45]M Chen. “Improved girault identification scheme”. In: *Electronics Letters* 30.19 (1994), pp. 1590–1591. URL: <https://doi.org/10.1049/el:19941106>.
- [46]H.-Y. Chien and C.-S. Lai. “ECC-based lightweight authentication protocol with untraceability for low-cost RFID”. In: *Journal of Parallel and Distributed Computing* 69.10 (2009), pp. 848–853. URL: <https://doi.org/10.1016/j.jpdc.2009.07.007>.
- [47]N. Chikouche, F. Cherif, P.-L. Cayrel, and M. Benmohammed. “A secure code-based authentication scheme for RFID systems”. In: *International Journal of Computer Network and Information Security* 7.9 (2015), p. 1. URL: <http://dx.doi.org/10.5815/ijcnis.2015.09.01>.

References

- [48]N. T. Courtois, M. Finiasz, and N. Sendrier. “How to Achieve a McEliece-Based Digital Signature Scheme”. In: *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13*. Vol. 2248. Lecture Notes in Computer Science. Springer, 2001, pp. 157–174. URL: https://doi.org/10.1007/3-540-45682-1_10.
- [49]J. M. Couveignes. “Hard Homogeneous Spaces”. In: *IACR Cryptol. ePrint Arch.* 2006 (2006), p. 291. URL: <http://eprint.iacr.org/2006/291>.
- [50]Y. Cui, K. Kobara, K. Matsuura, and H. Imai. “Lightweight Privacy-Preserving Authentication Protocols Secure against Active Attack in an Asymmetric Way”. In: *IEICE Trans. Inf. Syst.* 91-D.5 (2008), pp. 1457–1465. URL: <https://doi.org/10.1093/ietisy/e91-d.5.1457>.
- [51]J. Daemen and V. Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002. URL: <https://doi.org/10.1007/978-3-662-04722-4>.
- [52]B. D. Deebak, F. Al-Turjman, and L. Mostarda. “A Hash-Based RFID Authentication Mechanism for Context-Aware Management in IoT-Based Multimedia Systems”. In: *Sensors* 19.18 (2019), p. 3821. URL: <https://doi.org/10.3390/s19183821>.
- [53]D. Derler, H. C. Pöhls, K. Samelin, and D. Slamanig. “A General Framework for Redactable Signatures and New Constructions”. In: *ICISC 2015 - Seoul, South Korea*. Vol. 9558. Lecture Notes in Computer Science. Springer, 2015, pp. 3–19. URL: https://doi.org/10.1007/978-3-319-30840-1_1.
- [54]D. Derler, S. Ramacher, and D. Slamanig. “Post-Quantum Zero-Knowledge Proofs for Accumulators with Applications to Ring Signatures from Symmetric-Key Primitives”. In: *PQCrypto 2018, Fort Lauderdale, FL, USA*. Vol. 10786. Lecture Notes in Computer Science. Springer, 2018, pp. 419–440. URL: https://doi.org/10.1007/978-3-319-79063-3_20.
- [55]W. Diffie and M. E. Hellman. “New directions in cryptography”. In: *IEEE Trans. Inf. Theory* 22.6 (1976), pp. 644–654. URL: <https://doi.org/10.1109/TIT.1976.1055638>.
- [56]J. Ding and B.-Y. Yang. “Multivariate public key cryptography”. In: *Post-quantum cryptography*. Springer, 2009, pp. 193–241. URL: https://doi.org/10.1007/978-3-540-88702-7_6.
- [57]N. M. Döttling. “Cryptography based on the Hardness of Decoding”. PhD thesis. Karlsruhe Institute of Technology, 2014. URL: <https://crypto.itk.kit.edu/fileadmin/User/Doettling/thesis.pdf>.
- [58]M. F. Ezerman, H. T. Lee, S. Ling, K. Nguyen, and H. Wang. “A Provably Secure Group Signature Scheme from Code-Based Assumptions”. In: *Advances in Cryptology - ASIACRYPT 2015, Auckland, New Zealand, November 29 - December 3, 2015*. Vol. 9452. Lecture Notes in Computer Science. Springer, 2015, pp. 260–285. URL: https://doi.org/10.1007/978-3-662-48797-6_12.

- [59]J. Faugère, A. Otmani, L. Perret, and J. Tillich. “Algebraic Cryptanalysis of McEliece Variants with Compact Keys”. In: *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3*. Vol. 6110. Lecture Notes in Computer Science. Springer, 2010, pp. 279–298. URL: https://doi.org/10.1007/978-3-642-13190-5_14.
- [60]H. Feistel. “Cryptography and computer privacy”. In: *Scientific american* 228.5 (1973), pp. 15–23. URL: <http://www.jstor.org/stable/24923044>.
- [61]H. J. Fell and W. Diffie. “Analysis of a Public Key Approach Based on Polynomial Substitution”. In: *Advances in Cryptology - CRYPTO '85, Santa Barbara, California, USA, August 18-22*. Vol. 218. Lecture Notes in Computer Science. Springer, 1985, pp. 340–349. URL: https://doi.org/10.1007/3-540-39799-X_24.
- [62]A. Fiat and A. Shamir. “How to Prove Yourself: Practical Solutions to Identification and Signature Problems”. In: *CRYPTO '86, Santa Barbara, California, USA, 1986*. Vol. 263. Lecture Notes in Computer Science. Springer, 1986, pp. 186–194. URL: https://doi.org/10.1007/3-540-47721-7_12.
- [63]M. Finiasz, P. Gaborit, N. Sendrier, and S. Manuel. *SHA-3 proposal: FSB*. Proposal of a hash function for the NIST SHA-3 competition. 2008. URL: <https://hal.inria.fr/inria-00564012>.
- [64]P. Gaborit. “Shorter keys for code based cryptography”. In: *Proceedings of the 2005 International Workshop on Coding and Cryptography (WCC 2005)- Bergen, Norway*. ACM Press, 2005, pp. 81–91. URL: https://www.unilim.fr/pages_perso/philippe.gaborit/shortIC.ps.
- [65]P. Gaborit and M. Girault. “Lightweight code-based identification and signature”. In: *IEEE International Symposium on Information Theory, ISIT 2007, Nice, France, June 24-29, 2007*. IEEE, 2007, pp. 191–195. URL: <https://doi.org/10.1109/ISIT.2007.4557225>.
- [66]P. Gaborit, A. Hauteville, D. H. Phan, and J. Tillich. “Identity-Based Encryption from Codes with Rank Metric”. In: *CRYPTO, Santa Barbara, CA, USA*. Vol. 10403. Lecture Notes in Computer Science. Springer, 2017, pp. 194–224. URL: https://doi.org/10.1007/978-3-319-63697-9_7.
- [67]P. Gaborit, O. Ruatta, and J. Schrek. “On the Complexity of the Rank Syndrome Decoding Problem”. In: *IEEE Trans. Inf. Theory* 62.2 (2016), pp. 1006–1019. URL: <https://doi.org/10.1109/TIT.2015.2511786>.
- [68]P. Gaborit, O. Ruatta, J. Schrek, and G. Zémor. “New Results for Rank-Based Cryptography”. In: *AFRICACRYPT 2014, Marrakesh, Morocco, May 28-30*. Vol. 8469. Lecture Notes in Computer Science. Springer, 2014, pp. 1–12. URL: https://doi.org/10.1007/978-3-319-06734-6_1.

References

- [69]P. Gaborit, J. Schrek, and G. Zémor. “Full Cryptanalysis of the Chen Identification Protocol”. In: *PQCrypto 2011, Taipei, Taiwan, November 29 - December 2*. Vol. 7071. Lecture Notes in Computer Science. Springer, 2011, pp. 35–50. URL: https://doi.org/10.1007/978-3-642-25405-5_3.
- [70]P. Gaborit and G. Zémor. “On the Hardness of the Decoding and the Minimum Distance Problems for Rank Codes”. In: *IEEE Trans. Information Theory* 62.12 (2016), pp. 7245–7252. URL: <https://doi.org/10.1109/TIT.2016.2616127>.
- [71]C. Gentry. “Fully homomorphic encryption using ideal lattices”. In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*. ACM, 2009, pp. 169–178. URL: <https://doi.org/10.1145/1536414.1536440>.
- [72]O. Goldreich. “Zero-Knowledge twenty years after its invention”. In: *Electronic Colloquium on Computational Complexity (ECCC)* 063 (2002). URL: <http://eccc.hpi-web.de/eccc-reports/2002/TR02-063/index.html>.
- [73]O. Goldreich and L. A. Levin. “A Hard-Core Predicate for all One-Way Functions”. In: *21st Annual ACM Symposium on Theory of Computing, Seattle, Washington, USA*. ACM, 1989, pp. 25–32. URL: <https://doi.org/10.1145/73007.73010>.
- [74]S. Goldwasser, S. Micali, and C. Rackoff. “The Knowledge Complexity of Interactive Proof Systems”. In: *SIAM J. Comput.* 18.1 (1989), pp. 186–208. URL: <https://doi.org/10.1137/0218012>.
- [75]S. D. Gordon, J. Katz, and V. Vaikuntanathan. “A Group Signature Scheme from Lattice Assumptions”. In: *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9*. Vol. 6477. Lecture Notes in Computer Science. Springer, 2010, pp. 395–412. URL: https://doi.org/10.1007/978-3-642-17373-8_23.
- [76]Q. Guo, T. Johansson, and P. Stankovski. “A Key Recovery Attack on MDPC with CCA Security Using Decoding Errors”. In: *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*. Vol. 10031. Lecture Notes in Computer Science. 2016, pp. 789–815. URL: https://doi.org/10.1007/978-3-662-53887-6_29.
- [77]Q. Guo, T. Johansson, and P. S. Wagner. “A Key Recovery Reaction Attack on QC-MDPC”. In: *IEEE Trans. Inf. Theory* 65.3 (2019), pp. 1845–1861. URL: <https://doi.org/10.1109/TIT.2018.2877458>.
- [78]R. W. Hamming. “Error detecting and error correcting codes”. In: *The Bell System Technical Journal* 29.2 (1950), pp. 147–160. URL: <https://doi.org/10.1002/j.1538-7305.1950.tb00463.x>.

- [79]A. Hauteville and J. Tillich. “New algorithms for decoding in the rank metric and an attack on the LRPC cryptosystem”. In: *IEEE International Symposium on Information Theory, ISIT 2015, Hong Kong, China, June 14-19, 2015*. IEEE, 2015, pp. 2747–2751. URL: <https://doi.org/10.1109/ISIT.2015.7282956>.
- [80]J. Hoffstein, J. Pipher, and J. H. Silverman. “NTRU: A Ring-Based Public Key Cryptosystem”. In: *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25*. Vol. 1423. Lecture Notes in Computer Science. Springer, 1998, pp. 267–288. URL: <https://doi.org/10.1007/BFb0054868>.
- [81]T. Huang, Y. Ding, Z. Wang, and H. Wang. “Ultralightweight RFID Authentication Protocol Based on Permutation Matrix Encryption”. In: *CyberC 2019, Guilin, China*. IEEE, 2019, pp. 417–423. URL: <https://doi.org/10.1109/CyberC.2019.00078>.
- [82]C. Hung-Yu. “Secure access control schemes for RFID systems with anonymity”. In: *7th International Conference on Mobile Data Management (MDM), Nara, Japan*. IEEE Computer Society, 2006, pp. 10–12. URL: <https://doi.org/10.1109/MDM.2006.137>.
- [83]M. Jakobsson, F. T. Leighton, S. Micali, and M. Szydlo. “Fractal Merkle Tree Representation and Traversal”. In: *CT-RSA 2003, San Francisco, CA, USA, April 13-17*. Vol. 2612. Lecture Notes in Computer Science. Springer, 2003, pp. 314–326. URL: https://doi.org/10.1007/3-540-36563-X_21.
- [84]D. Jao and L. D. Feo. “Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies”. In: *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2*. Vol. 7071. Lecture Notes in Computer Science. Springer, 2011, pp. 19–34. URL: https://doi.org/10.1007/978-3-642-25405-5_2.
- [85]M. P. Jhanwar and R. Safavi-Naini. “Compact Accumulator Using Lattices”. In: *SPACE 2015, Jaipur, India*. Vol. 9354. Lecture Notes in Computer Science. Springer, 2015, pp. 347–358. URL: https://doi.org/10.1007/978-3-319-24126-5_20.
- [86]A. Kiayias and M. Yung. “Extracting Group Signatures from Traitor Tracing Schemes”. In: *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8*. Vol. 2656. Lecture Notes in Computer Science. Springer, 2003, pp. 630–648. URL: https://doi.org/10.1007/3-540-39200-9_39.
- [87]A. Kiayias and M. Yung. “Group Signatures with Efficient Concurrent Join”. In: *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26*. Vol. 3494. Lecture Notes in Computer Science. Springer, 2005, pp. 198–214. URL: https://doi.org/10.1007/11426639_12.
- [88]A. Kumar, P. Lafourcade, and C. Lauradoux. “Performances of cryptographic accumulators”. In: *IEEE, Local Computer Networks LCN 2014, Edmonton, AB, Canada*. IEEE Computer Society, 2014, pp. 366–369. URL: <https://doi.org/10.1109/LCN.2014.6925793>.

References

- [89]F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé. “Lattice-Based Group Signatures with Logarithmic Signature Size”. In: *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II*. Vol. 8270. Lecture Notes in Computer Science. Springer, 2013, pp. 41–61. URL: https://doi.org/10.1007/978-3-642-42045-0_3.
- [90]A. Langlois, S. Ling, K. Nguyen, and H. Wang. “Lattice-Based Group Signature Scheme with Verifier-Local Revocation”. In: *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28*. Vol. 8383. Lecture Notes in Computer Science. Springer, 2014, pp. 345–361. URL: https://doi.org/10.1007/978-3-642-54631-0_20.
- [91]J. Li, N. Li, and R. Xue. “Universal Accumulators with Efficient Nonmembership Proofs”. In: *ACNS 2007, Zhuhai, China*. Vol. 4521. Lecture Notes in Computer Science. Springer, 2007, pp. 253–269. URL: https://doi.org/10.1007/978-3-540-72738-5_17.
- [92]Z. Li, R. Zhang, Y. Yang, and Z. Li. “A provable secure mutual RFID authentication protocol based on error-correct code”. In: *CyberC, Shanghai, China*. IEEE, 2014, pp. 73–78. URL: <https://doi.org/10.1109/CyberC.2014.21>.
- [93]B. Libert, S. Ling, K. Nguyen, and H. Wang. “Zero-Knowledge Arguments for Lattice-Based Accumulators: Logarithmic-Size Ring Signatures and Group Signatures Without Trapdoors”. In: *EUROCRYPT 2016, Vienna, Austria*. Vol. 9666. Lecture Notes in Computer Science. Springer, 2017, pp. 1–31. URL: https://doi.org/10.1007/978-3-662-49896-5_1.
- [94]B. Libert, T. Peters, and M. Yung. “Scalable Group Signatures with Revocation”. In: *EUROCRYPT 2012, Cambridge, UK, April 15-19*. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 609–627. URL: https://doi.org/10.1007/978-3-642-29011-4_36.
- [95]S. Ling, K. Nguyen, and H. Wang. “Group Signatures from Lattices: Simpler, Tighter, Shorter, Ring-Based”. In: *Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1*. Vol. 9020. Lecture Notes in Computer Science. Springer, 2015, pp. 427–449. URL: https://doi.org/10.1007/978-3-662-46447-2_19.
- [96]S. Ling, K. Nguyen, H. Wang, and Y. Xu. “Lattice-based group signatures: Achieving full dynam-icity (and deniability) with ease”. In: *Theor. Comput. Sci.* 783 (2019), pp. 71–94. URL: <https://doi.org/10.1016/j.tcs.2019.03.023>.
- [97]S. Ling, K. Nguyen, H. Wang, and Y. Xu. “Lattice-Based Group Signatures: Achieving Full Dy-namicity with Ease”. In: *ACNS 2017, Kanazawa, Japan*. Vol. 10355. Lecture Notes in Computer Science. Springer, 2017, pp. 293–312. URL: https://doi.org/10.1007/978-3-319-61204-1_15.

- [98]Z. Liu, W. Zhang, and C. Wu. “A Lightweight Code-Based Authentication Protocol for RFID Systems”. In: *Applications and Techniques in Information Security*. Vol. 557. Communications in Computer and Information Science. Springer, 2015, pp. 114–128. URL: https://doi.org/10.1007/978-3-662-48683-2_11.
- [99]P. Loidreau. “Properties of codes in rank metric”. In: *CoRR* abs/cs/0610057 (2006). arXiv: cs/0610057. URL: <http://arxiv.org/abs/cs/0610057>.
- [100]B. Malek and A. Miri. “Lightweight mutual RFID authentication”. In: *ICC, Ottawa, Canada*. IEEE, 2012, pp. 868–872. URL: <https://doi.org/10.1109/ICC.2012.6363871>.
- [101]K. Mansoor, A. Ghani, S. A. Chaudhry, S. Shamshirband, S. Ghayyur, and A. Mosavi. “Securing IoT-Based RFID Systems: A Robust Authentication Protocol Using Symmetric Cryptography”. In: *Sensors* 19.21 (2019), p. 4752. URL: <https://doi.org/10.3390/s19214752>.
- [102]T. Matsumoto and H. Imai. “Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption”. In: *Advances in Cryptology — EUROCRYPT ’88*. Ed. by D. Barstow, W. Brauer, P. Brinch Hansen, D. Gries, D. Luckham, C. Moler, A. Pnueli, G. Seegmüller, J. Stoer, N. Wirth, and C. G. Günther. Vol. 330. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 419–453. URL: https://doi.org/10.1007/3-540-45961-8_39.
- [103]R. J. McEliece. “A public-key cryptosystem based on algebraic”. In: *Coding Thv* 4244 (1978), pp. 114–116. URL: https://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF.
- [104]C. A. Melchor, P. Cayrel, P. Gaborit, and F. Laguillaumie. “A New Efficient Threshold Ring Signature Scheme Based on Coding Theory”. In: *IEEE Trans. Information Theory* 57.7 (2011), pp. 4833–4842. URL: <https://doi.org/10.1109/TIT.2011.2145950>.
- [105]R. Misoczki, J. Tillich, N. Sendrier, and P. S. L. M. Barreto. “MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes”. In: *IEEE International Symposium on Information Theory, Istanbul, Turkey, July 7-12*. IEEE, 2013, pp. 2069–2073. URL: <https://doi.org/10.1109/ISIT.2013.6620590>.
- [106]M. Naveed, W. Habib, U. Masud, U. Ullah, and G. Ahmad. “Reliable and Low Cost RFID Based Authentication System for Large Scale Deployment.” In: *IJ Network Security* 14.3 (2012), pp. 173–179. URL: <http://ijns.femto.com.tw/contents/ijns-v14-n3/ijns-2012-v14-n3-p173-179.pdf>.
- [107]K. Nguyen, H. Tang, H. Wang, and N. Zeng. “New Code-Based Privacy-Preserving Cryptographic Constructions”. In: *Advances in Cryptology - ASIACRYPT 2019 -Kobe, Japan, December 8-12*. Ed. by S. D. Galbraith and S. Moriai. Vol. 11922. Lecture Notes in Computer Science. Springer, 2019, pp. 25–55. URL: https://doi.org/10.1007/978-3-030-34621-8_2.
- [108]L. Nguyen and R. Safavi-Naini. “Efficient and Provably Secure Trapdoor-Free Group Signature Schemes from Bilinear Pairings”. In: *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9*. Vol. 3329. Lecture Notes in Computer Science. Springer, 2004, pp. 372–386. URL: https://doi.org/10.1007/978-3-540-30539-2_26.

References

- [109]R. Nojima, H. Imai, K. Kobara, and K. Morozov. “Semantic security for the McEliece cryptosystem without random oracles”. In: *Designs, Codes and Cryptography* 49.1 (2008), pp. 289–305. URL: <http://dx.doi.org/10.1007/s10623-008-9175-9>.
- [110]K. Ouafi and R. C. W. Phan. “Privacy of Recent RFID Authentication Protocols”. In: *ISPEC, Sydney, Australia*. Vol. 4991. Lecture Notes in Computer Science. Springer, 2008, pp. 263–277. URL: https://doi.org/10.1007/978-3-540-79104-1_19.
- [111]D. Pointcheval and J. Stern. “Security Arguments for Digital Signatures and Blind Signatures”. In: *J. Cryptology* 13.3 (2000), pp. 361–396. URL: <https://doi.org/10.1007/s001450010003>.
- [112]D. Pointcheval and J. Stern. “Security Arguments for Digital Signatures and Blind Signatures”. In: *J. Cryptol.* 13.3 (2000), pp. 361–396. URL: <https://doi.org/10.1007/s001450010003>.
- [113]J. M. Pollard. “Factoring with cubic integers”. In: *The development of the number field sieve*. Ed. by A. K. Lenstra and H. W. Lenstra. Vol. 1554. Lecture Notes in Mathematics. Springer Berlin Heidelberg, 1993, pp. 4–10. URL: <https://doi.org/10.1007/BFb0091536>.
- [114]O. Regev. “On lattices, learning with errors, random linear codes, and cryptography”. In: *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*. ACM, 2005, pp. 84–93. URL: <https://doi.org/10.1145/1060590.1060603>.
- [115]R. L. Rivest, A. Shamir, and L. M. Adleman. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”. In: *Commun. ACM* 21.2 (1978), pp. 120–126. URL: <http://doi.acm.org/10.1145/359340.359342>.
- [116]A. Rostovtsev and A. Stolbunov. “Public-Key Cryptosystem Based on Isogenies”. In: *IACR Cryptol. ePrint Arch.* 2006 (2006), p. 145. URL: <http://eprint.iacr.org/2006/145>.
- [117]B. Schneier. *Applied cryptography - protocols, algorithms, and source code in C, 2nd Edition*. Wiley, 1996. URL: <http://www.worldcat.org/oclc/32311687>.
- [118]A. Shamir. “An Efficient Identification Scheme Based on Permuted Kernels (Extended Abstract)”. In: *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24*. Vol. 435. Lecture Notes in Computer Science. Springer, 1989, pp. 606–609. URL: https://doi.org/10.1007/0-387-34805-0_54.
- [119]C. E. Shannon. “A mathematical theory of communication”. In: *Bell Syst. Tech. J.* 27.3 (1948), pp. 379–423. URL: <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>.
- [120]P. W. Shor. “Algorithms for Quantum Computation: Discrete Logarithms and Factoring”. In: *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA*. IEEE Computer Society, 1994, pp. 124–134. URL: <https://doi.org/10.1109/SFCS.1994.365700>.
- [121]P. W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM J. Comput.* 26.5 (1997), pp. 1484–1509. URL: <https://doi.org/10.1137/S0097539795293172>.

- [122]M. Sidorov, M. T. Ong, R. V. Sridharan, J. Nakamura, R. Ohmura, and J. H. Khor. “Ultralightweight Mutual Authentication RFID Protocol for Blockchain Enabled Supply Chains”. In: *IEEE Access* 7 (2019), pp. 7273–7285. URL: <https://doi.org/10.1109/ACCESS.2018.2890389>.
- [123]J. Stern. “A method for finding codewords of small weight”. In: *Coding Theory and Applications, 3rd International Colloquium, Toulon, France, November 2-4*. Vol. 388. Lecture Notes in Computer Science. Springer, 1988, pp. 106–113. URL: <https://doi.org/10.1007/BFb0019850>.
- [124]J. Stern. “A new paradigm for public key identification”. In: *Information Theory, IEEE Transactions on* 42.6 (1996), pp. 1757–1768. URL: <https://doi.org/10.1109/18.556672>.
- [125]J. Stern. “Designing Identification Schemes with Keys of Short Size”. In: *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25*. Vol. 839. Lecture Notes in Computer Science. Springer, 1994, pp. 164–173. URL: https://doi.org/10.1007/3-540-48658-5_18.
- [126]The FlexiProvider group at Technische Universität Darmstadt. *FlexiProvider, an open source Java Cryptographic Service Provider*. 2001-2018, <http://www.flexiprovider.de/>, visited on 01/05/2019. URL: <http://www.flexiprovider.de/>.
- [127]P. Véron. “Improved identification schemes based on error-correcting codes”. In: *Appl. Algebra Eng. Commun. Comput.* 8.1 (1996), pp. 57–69. URL: <https://doi.org/10.1007/s002000050053>.
- [128]Z. Yu, M. H. Au, R. Yang, J. Lai, and Q. Xu. “Lattice-Based Universal Accumulator with Nonmembership Arguments”. In: *ACISP 2018, Wollongong, NSW, Australia*. Vol. 10946. Lecture Notes in Computer Science. Springer, 2018, pp. 502–519. URL: https://doi.org/10.1007/978-3-319-93638-3_29.
- [129]X. Zhang and J. Lu. “A new lightweight RFID mutual authentication protocol based on CRC code”. In: *IJICT* 13.4 (2018), pp. 490–501. URL: <https://doi.org/10.3390/s18030760>.