

THESIS

*Submitted in fulfillment of the requirements for the degree of
Philosophy Doctor (Ph.D)*

Research Center: Energy Research Center

Research Structure: Modeling and Simulation in Mechanics and Energetics Team

Discipline: Mathematics-Computer Science

Specialty: Cryptography

Presented and Defended on 07/01/2019 By:

Abdenaby LAMIRI

***BLOCKCHAIN TECHNOLOGY
BITCOIN SECURITY RISK ANALYSIS***

JURY

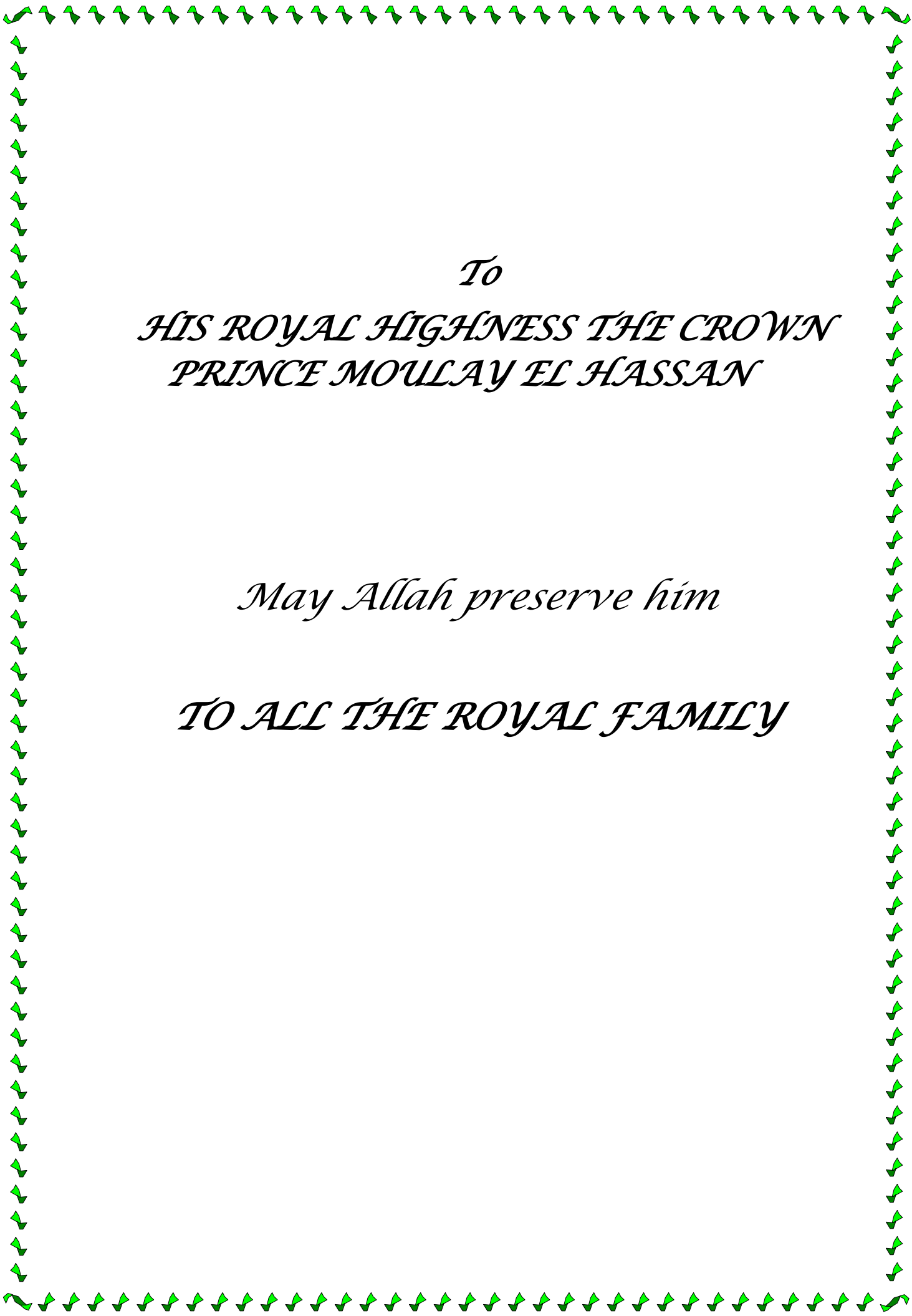
Gamal ZEGGWAGH,	PES, Faculty of Sciences, Mohammed V University, Rabat	Chair
Kamal GUERAOUI,	PES, Faculty of Sciences, Mohammed V University, Rabat	Reviewer
Bennasser BAHRAR,	PES, ENSET, Hassan II University, Casablanca	Reviewer
Mohamed O. BENSALAH,	PES, Faculty of Sciences, Mohammed V University,	Examiner
Ahmed MZERD,	PES, Faculty of Sciences, Mohammed V University, Rabat	Examiner



DEDICATIONS

*To
His Majesty Mohammed VI, King of
Morocco,
Supreme Chief and Chief of General
Staff of The Royal Armed Forces.*


*May Allah glorify him and preserve
his kingdom*




To
HIS ROYAL HIGHNESS THE CROWN
PRINCE MOULAY EL HASSAN

May Allah preserve him

TO ALL THE ROYAL FAMILY



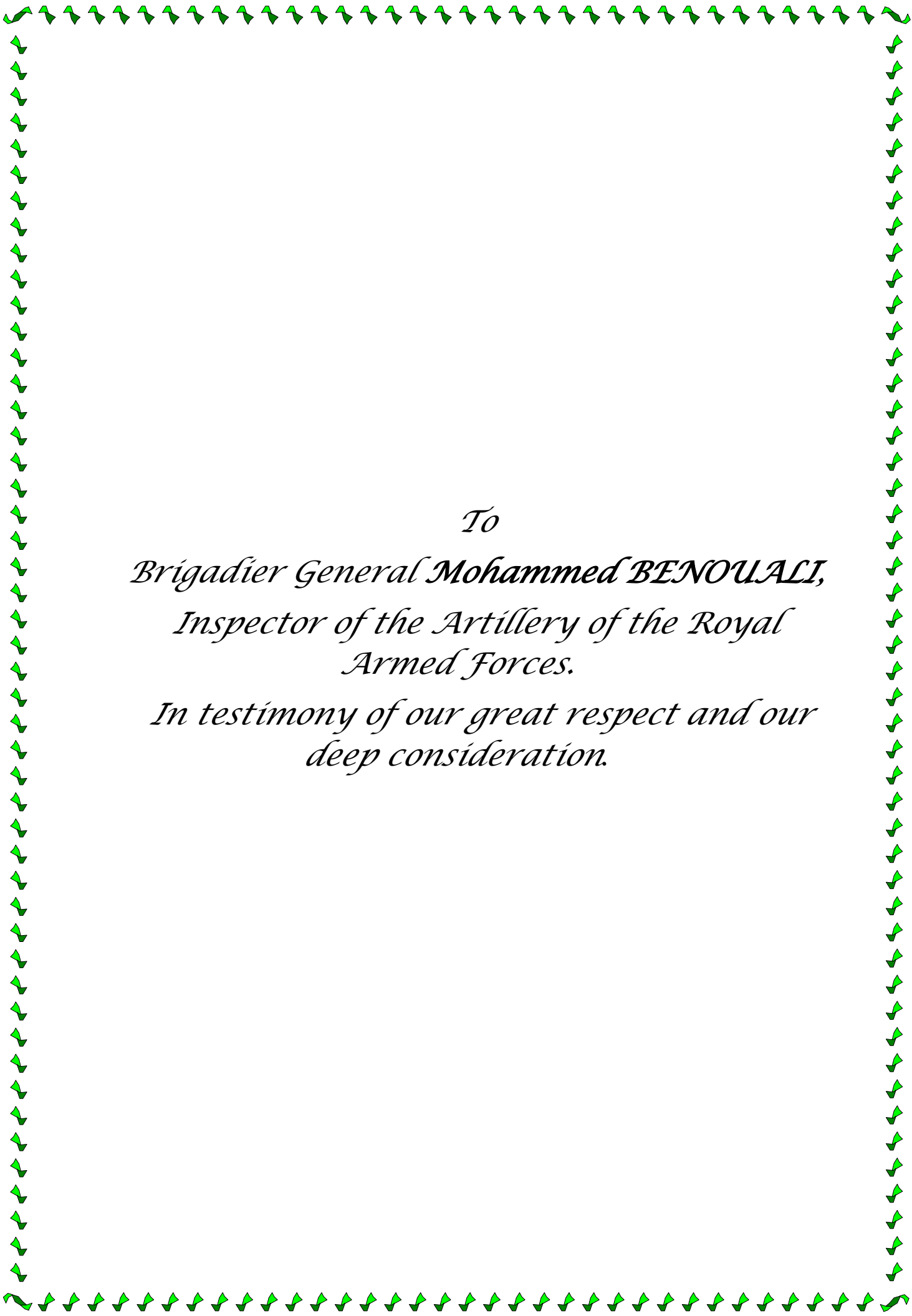
To
Lieutenant General Abdelfattah
LOUARAK,
Inspector General of the Royal Armed
Forces.
In testimony of our great respect and
our deep consideration.



To
Major General Mohammed BERRID,
Chief of 3rd Bureau of the General Staff
of the Royal Armed Forces.
In testimony of our great respect and
our deep consideration.

To
Brigadier General Abdeslam OTMANI,
Chief of 5th Bureau of the General Staff
of the Royal Armed Forces.

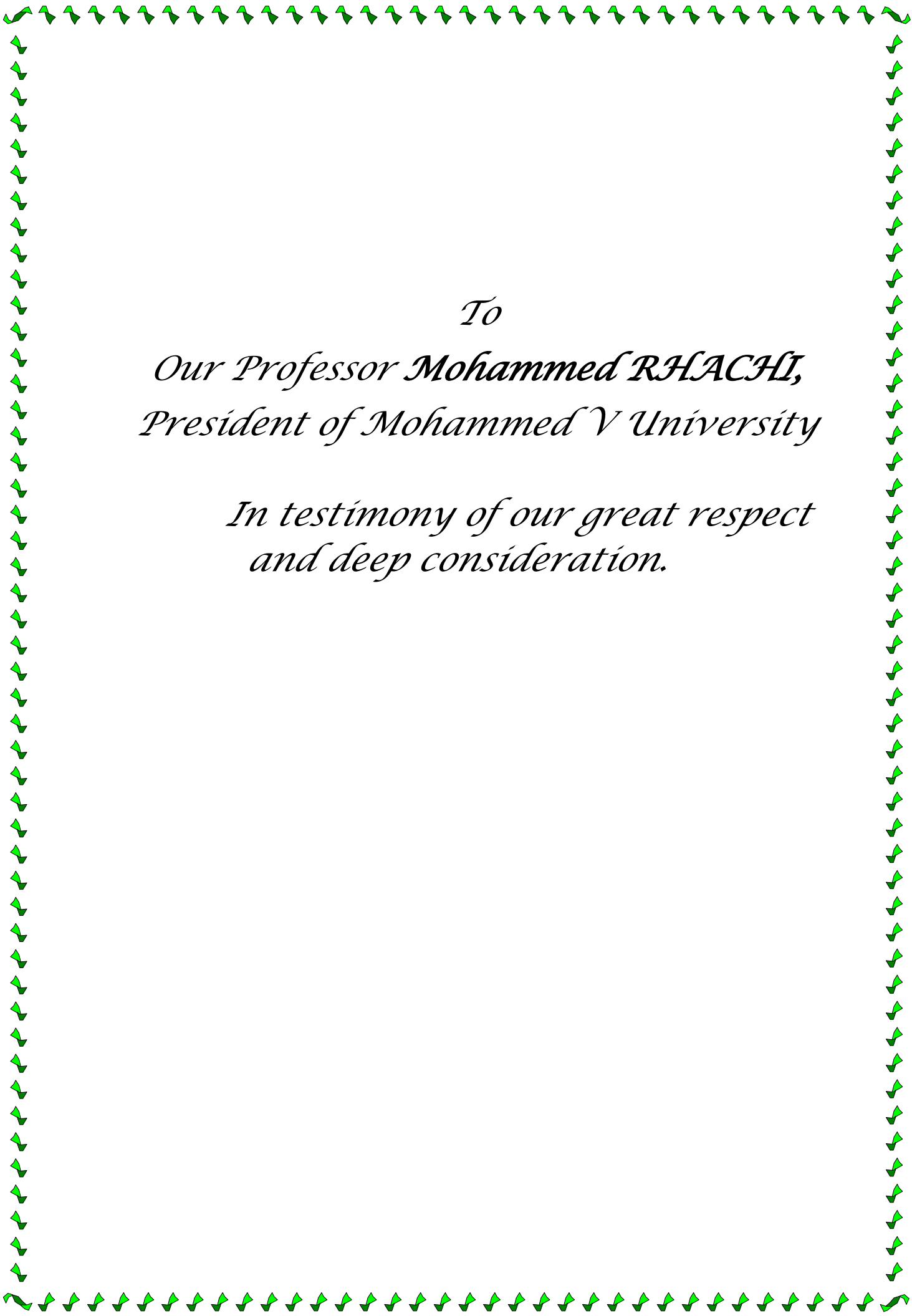
In testimony of our great respect and
our deep consideration.



To
Brigadier General Mohammed BENOUALI,
Inspector of the Artillery of the Royal
Armed Forces.
In testimony of our great respect and our
deep consideration.

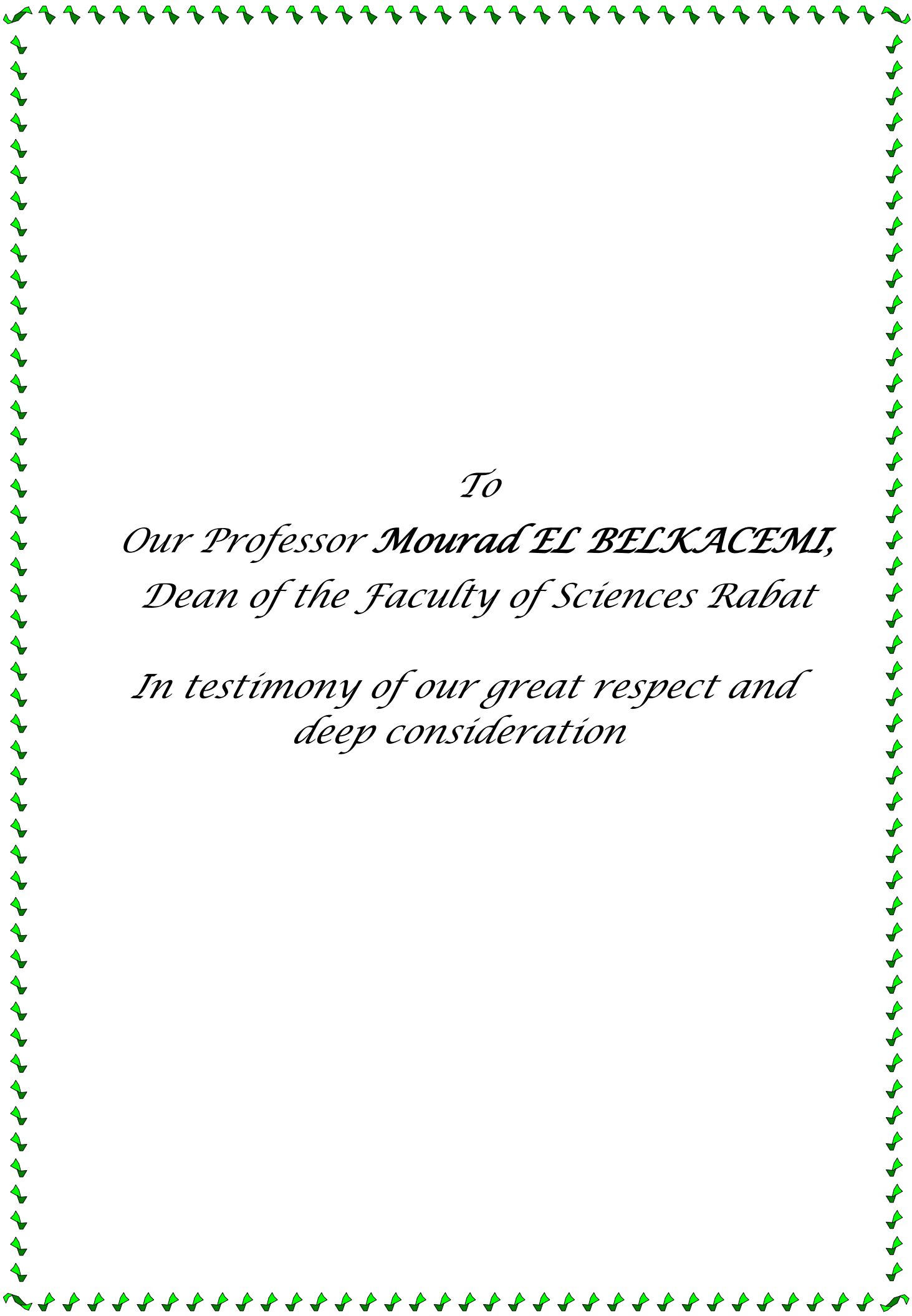
To
Brigadier General Hamid CHELI,
Colonel-Major Hassan NOUH,
Colonel-Major Hamid HLILA

*In testimony of our great respect and
our deep consideration for their guidance
and support.*



To
Our Professor Mohammed RHACHI,
President of Mohammed V University

In testimony of our great respect
and deep consideration.



To
Our Professor Mourad EL BELKACEMI,
Dean of the Faculty of Sciences Rabat

In testimony of our great respect and
deep consideration

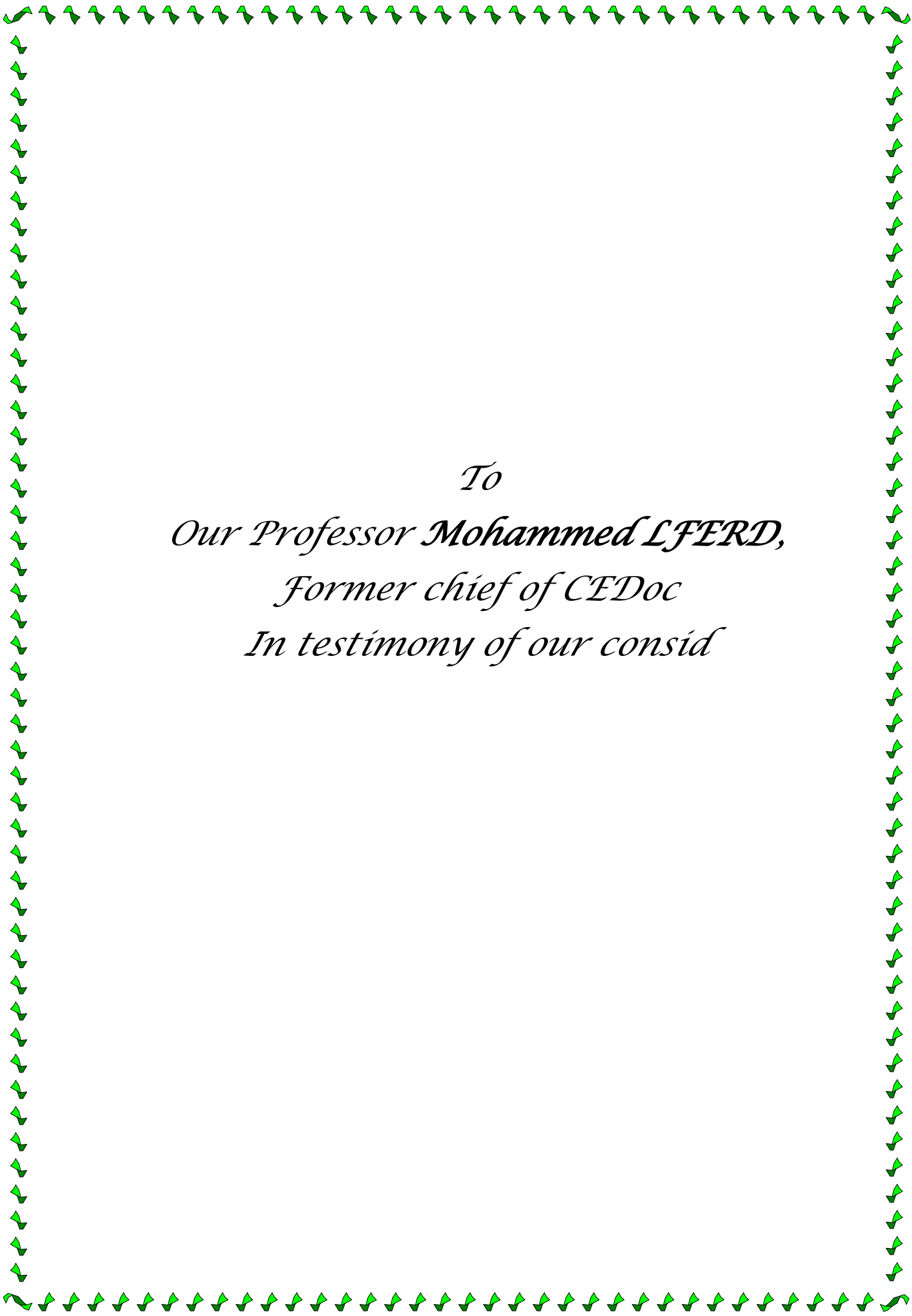
To
Our Master Professor Mohamed
HALIM,

Director of the Center for Doctoral
Studies "Science and Technology",


Faculty of Sciences-University
Mohammed V,

You have been, throughout this work, a
guide whose scientific and human
qualities, and availability have no equal
than.


You have always welcomed us with
kindness and sympathy.



To
Our Professor Mohammed LFERD,
Former chief of CEDoc
In testimony of our consid



To
Karima and Fadoua,
For their unwavering support
throughout this research journey.
Please accept my sincere consideration
and gratitude.



*To
Those who are dear to me
To
Those who have always believed in me
To
Those who have always encouraged me
I dedicate this thesis.*

ACKNOWLEDGMENTS

To
Our President of thesis Professor **Gamal ZEGGWAGH**,
Professor at the Faculty of Sciences of Rabat

We are very appreciative of the honor you give us by accepting the chairmanship of our thesis jury. Your scientific culture, your competence and your human qualities have inspired us with great admiration, and are an example for your students to follow. During our training, we had the privilege to benefit from your teaching and to appreciate your professional sense.

Please accept, dear Professor, the assurance of our esteem and our deep respect.

To
Our thesis judge Professor **Ahmed MZERD**,
Professor at the Faculty of Sciences of Rabat

We had the privilege of working with your team and appreciating your qualities and values. Your seriousness, your competence and your sense of duty have marked us enormously. Please find here the expression of our respectful consideration and deep admiration for all your scientific and human qualities. This work is an opportunity for us to express our deep gratitude.

To
Our Thesis Judge Professor **Kamal GUERAOU**,
Professor at the Faculty of Sciences of Rabat

You have been a master piece for this thesis. Your guidance and advices were of great value to our work. Please accept this Master work, as a token for all that you have done, and our great respect and deep appreciation.

To
Our Thesis Judge Professor **Bennasser BAHRAR**,
Professor at the Mohammedia ENSET

We have the privilege and the honor to have you among the members of our jury. Please accept our thanks and admiration for your teaching skills and your competence.

To
Our Thesis Judge Professor **Mohammed Ouadi BENSALAH**,
Professor at the Faculty of Sciences of Rabat

You did us the honor to accept with great kindness to sit on our jury of thesis. Please accept our great respect and deep appreciation for your efforts.

Résumé

Résumé

Blockchain est une plate-forme décentralisée qui gère les transactions, le stockage de données et les contrats intelligents entre utilisateurs. Elle a été utilisée pour la première fois par Bitcoin en 2008 et est maintenant utilisée par plus de 2000 crypto-monnaies. Elle a promis d'assurer l'anonymat de ses utilisateurs et l'intégrité des données par le biais de constructions cryptographiques de pointe, ce qui a accru l'intérêt et les investissements des utilisateurs. Dans cette thèse, nous fournissons une vue d'ensemble de la crypto-monnaie, basée sur la technologie Blockchain, le Bitcoin. Nous décrivons certaines limitations, problèmes de sécurité et applications de cette technologie. Notre contribution consiste en une analyse des risques, liés à l'utilisation de cette technologie comme étant une crypto-monnaie et aussi un moyen de paiement en ligne, à l'aide d'une méthode française d'évaluation de risques appelée "EBIOS". Dans une autre contribution, nous avons démontré l'importance et la corrélation entre la difficulté et la sécurité du Bitcoin. En outre, nous avons également suggéré un moyen de dissuader le problème du «minage égoïste» en s'appuyant sur une solution au problème du «dîner des philosophes». De plus, nous avons effectué une analyse du centre de gravité de Bitcoin à l'aide d'une analyse des facteurs critiques de ce system.

Mots-clés : Bitcoin; Blockchain; Crypto-monnaie; EBIOS; Risques; Sécurité.

Abstract

Abstract

Blockchain is a decentralized platform that manages transactions, data storage, and smart contracts between users without relying on any third party for processing. It was first used by Bitcoin in 2008 and now is being used by more than 2000 crypto-currencies. It promised to ensure the anonymity of its users and the integrity of the data through cutting-edge cryptographic constructs, which increased the users' interest and investments. In this thesis, we provide an overview of the most prominent crypto-currency, relying on the Blockchain technology, the Bitcoin. We outline some limitations, security issues, and applications of this technology. Our contribution consists of an analysis of the security risks pertaining to the use of Bitcoin as a currency and also as a payment system using a French risk assessment method called "EBIOS". Also, we demonstrated the importance and correlation between the difficulty feature and Security of Bitcoin System. Moreover, we suggested also a way to deter the selfish mining issue using a solution of the dining philosophers' problem. In addition, we carried out a Bitcoin center of gravity analysis using a US Army framework based on critical factors analysis of Bitcoin.

Keywords: Bitcoin; Blockchain; Crypto-currency; EBIOS; Risks; Security

Résumé Détaillé En
Français

I. Aperçu général sur les limites, problèmes de sécurité et applications du Bitcoin

Ce travail de recherche a analysé les limites, les problèmes de sécurité relatifs au Bitcoin et aussi les champs de son utilisation. Elle a relevé les points suivants :

I.1 Les limites

I.1.1 Capacité du débit de transactions

Bitcoin présente une limite quant à la capacité du débit des transactions à traiter par seconde qui est de l'ordre de 7 transactions par seconde. Ceci représente un handicap majeur si cette monnaie devient une monnaie à large spectre d'utilisation. Cette contrainte est liée à la taille très petite du block qui est de 1MB. La communauté des programmeurs a eu un différend sur l'augmentation de cette taille pour augmenter le débit des transactions. Cette dispute a causé la création de 3 fourchettes de Bitcoin : Bitcoin Core, Bitcoin Classic et Bitcoin XT (devenu Bitcoin Cash en 2017).

I.1.2 Latence de confirmation des transactions

Les performances de Bitcoin sont aussi réduites à cause d'une latence dans la confirmation des transactions. En effet, les transactions sont confirmées dans un délai de 10 minutes ce qui pénalise les transactions rapides liées au commerce électronique. Cette contrainte est due au temps mis par les mineurs pour résoudre la preuve du travail (appelée Proof-of-work). Cette limitation a été résolue par d'autres crypto-monnaies clones de Bitcoin comme Litecoin qui a réduit cette latence à 2.5 minutes et Dogecoin à même une 1 minute. Bitcoin présente toujours cette lacune capacitaire.

I.1.3 Gaspillage d'énergie

Le choix d'un consensus décentralisé basé sur la preuve du travail requiert beaucoup de moyens de calcul qui consomment beaucoup d'électricité et produisent beaucoup de chaleur. Ce type de consensus coûte très cher aux mineurs. La communauté Bitcoin doit chercher un autre type de consensus moins coûteux en ressources.

I.2 Les problèmes de sécurité

I.2.1 Sécurité des transactions à zéro confirmation

Pour pouvoir dépenser une nouvelle transaction, son propriétaire doit attendre au moins six confirmations, ce qui prend environ une heure ou plus. Cette contrainte pourrait gêner les entreprises utilisant Bitcoin comme moyen de paiement. Pour surmonter cette limitation, Bitcoin exhorte ses utilisateurs à payer avec leurs transactions à confirmation zéro, qui sont des transactions qui n'ont pas encore été confirmées par les mineurs. Cette alternative pose un problème de sécurité puisqu'elle peut être utilisée pour acquérir des services ou des biens sans que les acheteurs aient à dépenser leurs bitcoins, problème couramment appelé attaques de double dépense.

Les transactions à confirmation zéro ne sont pas sécurisées, car les attaquants peuvent facilement organiser des attaques par double dépense. Ce type d'attaque a une probabilité de succès de presque 100% lorsque l'attaquant utilise un ou plusieurs nœuds auxiliaires ayant pour

rôle de diffuser son attaque vers un grand nombre de nœuds connectés. Pour cette raison, les transactions sans confirmation zéro ne doivent pas être acceptées directement par les fournisseurs.

Une technique d'atténuation permettant de contrer cette attaque consiste pour le fournisseur à envisager une période d'écoute de quelques secondes avant de livrer les marchandises à l'acheteur. Pendant cette période, le fournisseur est plus susceptible de détecter le problème de la double dépense en surveillant le réseau. Cependant, l'attaquant est toujours capable de contourner la technique de détection en retardant la transmission des transactions à double dépense de manière à dépasser la période d'écoute, ainsi qu'en augmentant le nombre de nœuds auxiliaires.

I.2.2 fourchettes du Blockchain (chaîne de blocs)

La sécurité du Blockchain pour Bitcoin et Ethereum est assurée par un mécanisme de consensus basé sur la preuve du travail, soutenu par des mineurs qui consacrent leur puissance de calcul à la création de nouveaux blocs. Lorsque deux blocs sont trouvés en même temps, la Blockchain est fourchue. Cette situation se produit plusieurs fois au cours de la même journée, mais elle est résolue de manière inhérente par le système, qui considère la chaîne la plus longue avec la difficulté majeure comme la version valide du Blockchain. Dans le cas de fourchettes, les transactions qui n'apparaissent pas dans la version valide du Blockchain sont ajoutées ultérieurement dans les blocs suivants. Lorsque les fourchettes ne peuvent pas être résolues automatiquement par le système, les développeurs Bitcoin peuvent forcer la chaîne aux dépens de l'autre. Cet effet de levier des développeurs remet en question la décentralisation du système Bitcoin.

Les fourchettes de Bitcoin peuvent être exploitées pour lancer des attaques par double dépense plus importantes sur des transactions à zéro confirmation. Dans la mesure où une chaîne est finalement considérée comme la version valide de l'historique, toutes les transactions incluses dans l'autre version de la chaîne seraient invalidées par les mineurs. Certaines de ces transactions seront incluses dans des blocs ultérieurs tandis que les transactions à double dépense ne seront jamais incluses. Bitcoin n'atténue pas ce problème en remboursant les personnes perdantes.

En 2013, Bitcoin a connu de graves difficultés lorsque les développeurs ont publié le client Bitcoin version 0.8 qui implémentait une base de données LevelDB au lieu d'une base de données BerkleyDB utilisée dans la version 0.7. Ce problème a été résolu par l'intervention de développeurs Bitcoin qui ont forcé la plus petite chaîne à être la version valide du Blockchain. Ce problème aurait pu être évité si les développeurs de Bitcoin avaient conçu le client 0.8 de Bitcoin en tenant compte de la compatibilité ascendante.

I.2.3 Malléabilité de la transaction

En Bitcoin, les transactions sont identifiées par le hachage de leurs données. Chaque fois que les données changent, l'identifiant de la transaction change également. En Bitcoin, la signature de la transaction, également appelée donnée témoin, qui permet de débloquent les fonds peut toujours être valide en dépit de légères modifications. Ce changement dans les données de témoin produit un nouvel identifiant pour la même transaction. Cette vulnérabilité est appelée malléabilité de la transaction.

Cette question a été étudiée par C. Decker et R. Wattenhofer, qui ont examiné des allégations selon lesquelles MtGox aurait perdu 850 000 bitcoins en raison d'attaques de malléabilité. Ils ont conclu qu'à peine 386 bitcoins auraient pu être volés à l'aide d'attaques de malléabilité de MtGox. Ils ont également mentionné que la malléabilité des transactions est un problème réel et doit être traité dans toute implémentation de client Bitcoin. M. Andrychowicz, S. Dziembowski, D. Malinowski et L. Mazurek ont étudié le même problème et ont suggéré un protocole de dépôt assorti d'un plan d'engagement assorti d'une durée permettant de créer une transaction de «remboursement» résiliente à la malléabilité afin d'éviter ce problème de sécurité.

Ce problème de sécurité a été résolu avec le protocole de ségrégation des témoins. Ce protocole définit une nouvelle structure appelée témoin, qui contient les scripts et les signatures dans des blocs séparément de la Merkle tree des transactions.

I.2.4 L'attaque des 51%

La sécurité de Bitcoin Blockchain est étroitement liée à une hypothèse sous-jacente, qui affirme que les mineurs resteront toujours honnêtes et travailleront pour la sécurité Bitcoin, car ils sont récompensés financièrement. Toutefois, si un pool de mineurs malveillants détient plus de 51% de la puissance de calcul du réseau, le réseau sera vulnérable à ce que l'on appelle l'attaque des 51%, l'attaque de la majorité ou l'exploitation du minage égoïste. Beikverdi et al ont affirmé que le bitcoin n'était pas décentralisé car le marché minier était contrôlé par quelques groupes de miniers et que ce problème augmentait le risque d'une attaque à 51%. Ils ont également affirmé qu'en 2014, Bitcoin n'était décentralisé qu'à 33%.

G. Karame et autres ont mentionné que si un adversaire dispose de plus de 50% de la puissance de calcul, il peut, en théorie, doubler les transactions, empêcher la confirmation des transactions, empêcher les mineurs honnêtes de créer des blocs valides de manière à invalider l'ensemble de la sécurité du réseau.

Ittay Eyal et Emin Sirer ont suggéré que les mineurs égoïstes qui détiennent plus de 33% de la puissance de hachage du réseau peuvent toujours acquérir une part importante du processus de minage. Ils ont également mentionné qu'une stratégie minage égoïste qui consiste pour un mineur qui n'annonce pas ses blocs extraits sur le réseau afin d'augmenter leurs revenus et laisse les autres mineurs perdre leur temps et leur puissance de calcul.

Pour dissuader les exploitations du minage égoïste, Ittay Eyal et Emin Sirer suggèrent une stratégie incitant les mineurs à diffuser tous les blocs reçus et à choisir au hasard un bloc à exploiter en cas de concurrence de deux blocs.

I.2.5 Le problème de double dépense

La double dépense se produit lorsque les mêmes fonds sont dépensés deux fois. Bitcoin avait promis de résoudre ce problème grâce à un consensus fondé sur la preuve du travail. En dépit de ce strict contrôle de sécurité, les attaques par double dépense ont plus de chances de réussir en cas de fourchette de Blockchain. De plus, les attaques par double dépense sur les paiements rapides réussissent avec une probabilité considérable et peuvent être montées à faible coût. G. Karame et al. ont examiné la question des paiements rapides et ont suggéré des contre-mesures pour les clients légers permettant de détecter les attaques par double dépense. Ces mesures ont été mises en œuvre dans la plupart des fourchettes Bitcoin. Un moyen intégré d'empêcher les doubles dépenses en Bitcoin est que le destinataire des fonds ne peut pas utiliser

les bitcoins reçus tant que six blocs n'ont pas été ajoutés en haut du bloc contenant la transaction du destinataire. Une autre manière est la vérification de transaction qui consiste à vérifier la signature, le format, la correction des champs, l'équilibre suffisant et les entrées qui n'ont pas été utilisées dans des transactions antérieures.

I.3 Les applications de Bitcoins

I.3.1 Stockage décentralisé

La technologie Blockchain permet à tout utilisateur de créer un système de stockage décentralisé capable d'assurer un niveau élevé de disponibilité des données stockées. Cependant, il n'est pas recommandé de stocker des données volumineuses, en particulier dans la Blockchain Bitcoin.

Pour stocker des données, l'utilisateur génère une clé privée (K), puis chiffre les données à l'aide de la clé privée pour produire un objet chiffré (OC). Il stocke l'objet chiffré dans n nœuds et note les URI, identificateurs de ressources uniformes, (U_1, U_2, \dots, U_n) de l'OC stocké dans chaque nœud. Après cela, il chiffre tous les URI en utilisant la même clé (K) pour produire des URI cryptés (EC_1, EC_2, \dots, EC_n). Enfin, il stocke les URI chiffrés (ECs : 1 à n) dans le Blockchain ainsi qu'un hachage de l'objet chiffré (Hash (OC)) pour un contrôle d'intégrité ultérieur. Une fois sa transaction confirmée, l'utilisateur est alors assuré que les informations ne seront jamais altérées. La figure 1 décrit ce processus.

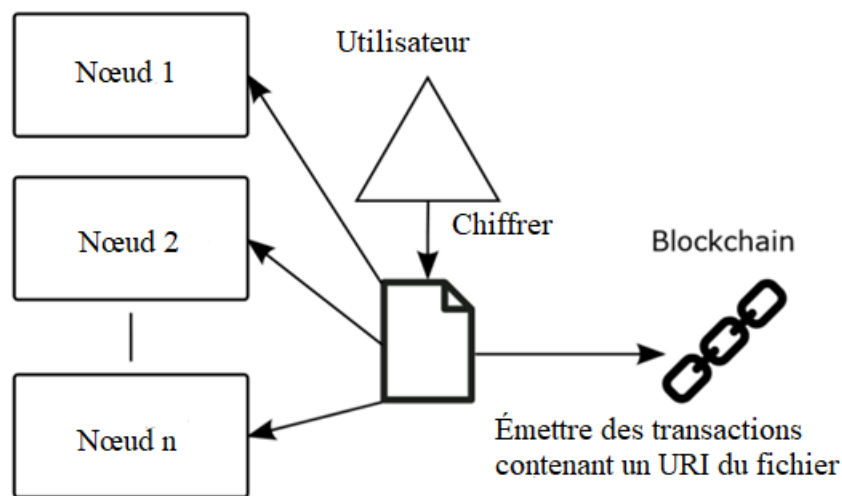


Figure.1 Stockage décentralisé en Bitcoin.

Pour récupérer les données stockées dans le Blockchain, l'utilisateur récupère les URIs chiffrés (UCs : de 1 à n), puis les déchiffre à l'aide de la clé privée (K). Il utilise les URIs pour récupérer les données stockées dans les n nœuds. Après cela, il vérifie l'intégrité à l'aide du hachage (OC) extrait du Blockchain et du hachage de l'objet chiffré stocké dans les nœuds. Une fois l'intégrité vérifiée, il déchiffre l'objet chiffré récupéré des nœuds à l'aide de la clé privée pour obtenir les données.

I.3.2 Gestion décentralisée de l'identité

La Blockchain permet également de construire un système de gestion d'identité décentralisé en enregistrant et en confirmant l'identité dans le Blockchain. Cette application

permet d'éviter l'usurpation d'identité et aide les personnes à conserver leur identité. Onecoin a implémenté cette application à l'aide d'un Blockchain dédié, appelé Onecoin Blockchain.

I.3.3 Contrats intelligents de Bitcoin

Les contrats intelligents sont comme des contrats réels. Ils se présentent sous la forme de petits programmes informatiques pouvant être stockés dans le Blockchain. Ces programmes sont exécutés une fois qu'un objectif ou une condition est atteint entre les deux parties qui les créent.

Contrairement à Ethereum, qui a été conçu avec un langage de programmation complet pouvant être utilisé pour créer des "contrats intelligents", Bitcoin a été implémenté avec un langage de script limité, ce qui n'a pas permis d'activer les contrats intelligents. Cependant, en utilisant certaines nouvelles fonctionnalités ajoutées à Bitcoin par le biais de propositions d'amélioration, certaines fonctionnalités de contrats intelligents peuvent être obtenues à l'aide de scripts Bitcoin. Le protocole d'amélioration de Bitcoin 65 (BIP65) a introduit un nouvel opcode, `OP_CHECKLOCKTIMEVERIFY`, considéré comme la fonctionnalité la plus importante pour les contrats intelligents en Bitcoin. Cet opcode permet d'écrire des scripts qui empêchent que les fonds d'un portefeuille à plusieurs signatures soient utilisés tant qu'un certain modèle de signature n'est pas mis en œuvre ou qu'un certain laps de temps ne s'écoule.

II. Problématique

L'objectif principal de cette thèse est d'analyser et évaluer les risques de sécurité relatifs à l'utilisation de Bitcoin. Pour ce faire, on a pensé qu'il était judicieux de transformer cet objectif en questions auxquelles on devrait apporter des réponses. Quatre questions principales ont été identifiées lors de la recherche bibliographique. Ces questions servent à la fois l'objectif de la thèse et apportent une contribution pour combler le gap dans la recherche pour ce sujet. Il s'agit des questions suivantes :

- Quels sont les principaux risques liés à l'utilisation de Bitcoin comme crypto-monnaie et système de paiement?
- Quelles sont les fonctionnalités de sécurité intégrées à Bitcoin et comment la difficulté contribue-t-elle à la sécurité du système?
- Comment pouvons-nous prévenir les attaques de la majorité ou l'exploitation du minage égoïste?
- Quel est le centre de gravité Bitcoin et que faut-il faire pour plus de sécurité?

Ces 4 questions ont fait l'objet de 4 articles dont deux ont été déjà publiés dans des journaux et conférences internationales. Chaque article a étudié une seule problématique et y a proposé une solution.

III. Contribution

III.1 Article I

III.1.1 Risques majeurs liés au Bitcoin

Pour répondre à la première question, nous avons utilisé une méthode qualitative d'analyse et d'évaluation des risques : EBIOS (Expression des besoins et identification des objectifs de sécurité). Dans le premier article nous avons relevé 12 risques majeurs auxquels Bitcoin devraient faire face. Ces risques sont :

- R0 - Risque lié à la disponibilité des transactions
- R1 - Risque lié à l'intégrité des transactions
- R2 - Risque lié à la confidentialité des transactions
- R3 - Risque lié à la disponibilité du bloc
- R4 - Risque lié à l'intégrité du bloc
- R5 - Risque lié à la disponibilité du Blockchain
- R6 - Risque lié à l'intégrité du Blockchain
- R7 - Risque lié à la disponibilité du mécanisme de consensus
- R8 - Risque lié à l'intégrité du mécanisme de consensus
- R9 - Risque lié à la disponibilité des clés privées
- R10 - Risque lié à l'intégrité des clés privées
- R11 - Risque lié à la confidentialité des clés privées

III.1.2 Évaluation des risques identifiés

Nous avons évalué ces risques selon leur importance en fonction de critères préétablis de gestion des risques. Certains risques ont été omis parce qu'ils étaient jugés faibles. La Figure.2 illustre l'évaluation des risques après la prise en compte des mesures de sécurité.

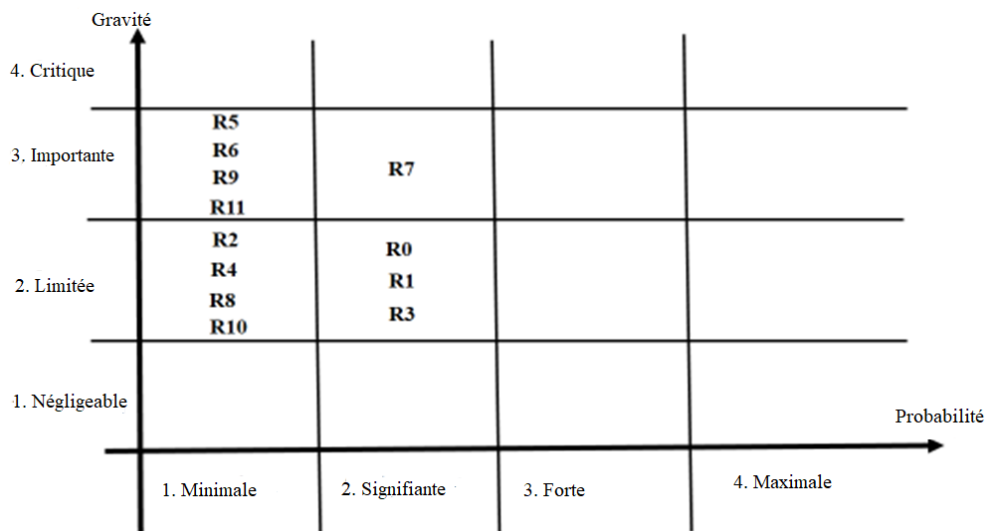


Figure.2. Illustration de l'évaluation des risques

Pour évaluer ces risques nous avons préétabli des critères de gestion de risques, qui distinguent entre 4 catégories de risques, qui sont :

- Risques intolérables: Ce sont ceux de gravité critique et ceux de gravité importante avec une probabilité forte ou maximale. Dans notre cas, nous n'avons aucun risque intolérable.

- Risques très importants: Ce sont ceux de gravité et de probabilité importantes, et ceux de gravité limitée et de probabilité forte ou maximale. Dans notre cas, R7 représente un risque très important.

- Risques importants: Ce sont ceux qui présentent une gravité importante et une probabilité minimale ou ceux qui sont d'une gravité limitée avec une probabilité significative. Dans notre cas, ces risques sont: R0, R1, R3 et R5, R6, R9, R11.

- Risques négligeables: Ce sont ceux dont la gravité est limitée et la probabilité minimale. Dans notre cas, ces risques sont R2, R4, R8 et R10.

III.1.3 Identification des objectifs de sécurité

L'analyse des risques liés à la sécurité de Bitcoin a révélé quatre types de risques qui devraient être traités en fonction des critères de gestion des risques établis. Nous devrions identifier les options pour traiter ces risques. Il existe quatre options pour traiter un risque: éviter (ou refuser), réduire, accepter, transférer (ou partager).

Selon les critères de gestion des risques retenus, les risques intolérables, très importants et importants doivent être réduits à un niveau acceptable, transférés ou évités si cela est possible. Des risques négligeables peuvent être acceptés. Par conséquent, les risques tels que R0, R1, R3, R5, R6, R7 et R9 doivent être réduits à un niveau acceptable, tandis que les risques tels que R2, R4, R8 et R10 peuvent être acceptés. R11 peut être réduit ou transféré à une tierce partie telle qu'une compagnie d'assurance pour partager le risque de perte ou de vol. Ces objectifs de sécurité sont illustrés dans le tableau I ci-dessous.

TABLEAU I : OBJECTIFS DE SECURITE POUR TRAITER LES RISQUES LIES A BITCOIN

Risque	Gravité	Probabilité	Objectif de sécurité
R0–Risque lié à la disponibilité des transactions	Limitée	Signifiante	Réduire
R1–Risque lié à l'intégrité des transactions	Limitée	Signifiante	Réduire
R2–Risque lié à la confidentialité des transactions	Limitée	Minimal	Accepter
R3–Risque lié à la disponibilité du bloc	Limitée	Signifiante	Réduire
R4–Risque lié à l'intégrité du bloc	Limitée	Minimale	Accepter
R5–Risque lié à la disponibilité du Blockchain	Importante	Minimale	Réduire
R6–Risque lié à l'intégrité du Blockchain	Importante	Minimale	Réduire
R7–Risque lié à la disponibilité du mécanisme de consensus	Importante	Signifiante	Réduire

R8–Risque lié à l'intégrité du mécanisme de consensus	Limitée	Minimale	Accepter
R9–Risque lié à la disponibilité des clés privées	Importante	Minimale	Réduire
R10–Risque lié à l'intégrité des clés privées	Limitée	Minimale	Accepter
R11–Risque lié à la confidentialité des clés privées	Importante	Minimale	Réduire ou Transférer

III.1.4 Gestion des risques résiduels

Après avoir atteint chaque objectif de sécurité, certains risques pourraient subsister et devraient également être traités. Ce sont les risques résiduels auxquels nous devons faire face avec des mesures de sécurité complémentaires. Pour cela, nous allons considérer les règles suivantes:

- o Les risques évités ne génèrent pas de risques résiduels s'ils étaient complètement évités.
- o Les risques réduits entraînent des risques résiduels s'ils ne sont pas complètement réduits
- o Les risques acceptés sont entièrement des risques résiduels
- o Les risques transférés n'impliquent pas de risques résiduels.

Ces risques résiduels qui subsisteront après la mise en œuvre des mesures de sécurité doivent également être estimés en termes de gravité et de probabilité. Le tableau II illustre les principaux risques résiduels ainsi que leur évaluation.

TABLEAU II RISQUES RÉSIDUELS

Risque résiduel	Gravité	Probabilité
Risque lié à la compromission du Blockchain	Importante	Minimale
Risque lié à la divulgation des clés privées	Importante	Minimale

III.1.5 Mesures de sécurité complémentaires

Nous avons proposé des mesures de sécurité adaptées de la norme ISO 27002. Ces mesures, si elles sont correctement mises en œuvre, contribueraient à réduire la gravité et la probabilité des risques les plus identifiés pour Bitcoin.

L'hypothèse sous-jacente de la sécurité Bitcoin était basée sur l'honnêteté des mineurs. Ces mineurs gagnent plus en honnêteté grâce à la récompense qu'ils reçoivent pour les nouveaux blocs minés et aux frais de transaction qu'ils perçoivent. Bien que cette hypothèse reste solide à l'avenir, les parties prenantes de Bitcoin devraient investir dans l'exploitation du minage afin de conserver la majorité de la puissance de hachage et, partant, d'assurer une plus grande sécurité à leurs bitcoins. Cette mesure, ajoutée à la surveillance fréquente et à l'évaluation des risques, réduirait à un niveau acceptable le risque de compromission de la Blockchain. En outre,

le transfert du risque lié à la divulgation des clés privées à une compagnie d'assurance réduirait certainement son impact et, par conséquent, maintiendrait ce risque à un niveau tolérable.

III.2 Article 2

Notre deuxième contribution, intitulée «Difficulté Bitcoin, une fonctionnalité de sécurité», répond à la deuxième question de cette thèse, qui est: «Quelles sont les fonctionnalités de sécurité intégrées à Bitcoin et comment la difficulté contribue-t-elle à la sécurité du système?»

Nous avons montré comment le Target Difficulté est calculé à partir du Bits

- $\text{Bits} = (2 \text{ bits Exposant})(4 \text{ bits représentent le coefficient})$

Exemple Bits= 1d00ffff

- $\text{TARGET} = \text{COEFFICIENT} * 2^{(8 * (\text{EXPONENT} - 3))}$

Nous avons développé un script en python qui calcule le Target à partir du Bits. Après nous avons présenté comment la difficulté est calculée :

- $\text{DIFFICULTÉ} = \text{TARGET DU 1}^\circ \text{ BLOCK} / \text{TARGET DU BLOCK COURANT}$

Nous avons développé un script en python qui calcule la difficulté. Nous avons aussi illustré la formule de retargetting qui se produit dynamiquement dans chaque nœud complet après chaque 2016 blocs (2 semaines environ)

$\text{NEW TARGET} = \text{CURRENT TARGET} * (\text{TIME ON MINUTES OF THE LAST 2016 BLOCKS}) / 20160 \text{ MINUTES}$

Nous avons aussi démontré la corrélation entre la difficulté et la puissance de calcul du réseau (Tableau III)

TABLEAU III : CORRÉLATION ENTRE LA DIFFICULTÉ ET LE TAUX DE HACHAGE DU RÉSEAU

Date	Difficulté	Taux de Hachage (GH/s)
06/12/2017	1,590,896,927,258	11,388,083,790
02/12/2016	286,765,766,821	2,052,749,317
Taux De Croissance	5.547722606133257	5.547722605818799

Nous avons déduit l'importance de la difficulté dans la sécurité du Blockchain.

III.3 Article 3

Notre troisième article, intitulé «Utilisation de la solution randomisée du problème du dîner des philosophes pour empêcher l'attaque de la majorité du Bitcoin», apporte une réponse à la troisième question, à savoir: «Comment pouvons-nous prévenir les attaques de la majorité ou l'exploitation du minage égoïste? »

Nous avons proposé une modification dans l'architecture du réseau de Bitcoin par l'ajout d'un nœud arbitre illustré dans la Figure.3.

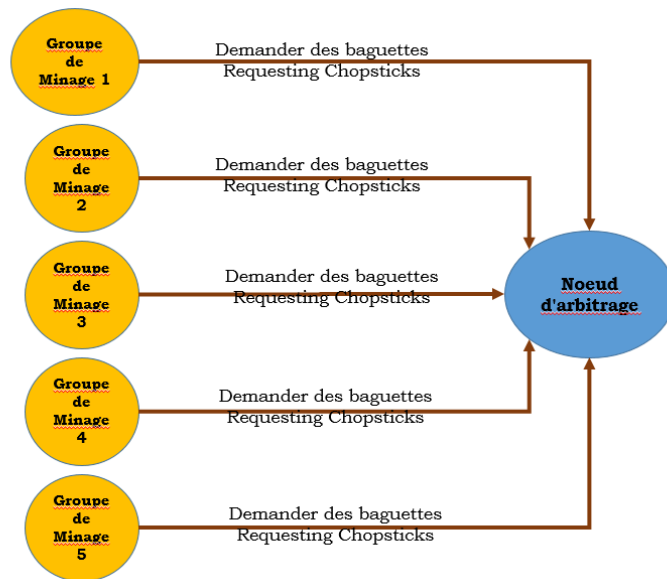


Figure.3. Nœud arbitre régulant le minage selon l’algorithme du Dîner des philosophes.

Le nœud d'arbitrage que nous proposons s'assurera que chaque pool minier aura une chance de créer un Bloc et que personne ne monopolisera le processus de minage. Le processus d'extraction se déroulera en deux phases: prendre le rôle de miner et miner. Alors que la première phase ajoute une certaine centralisation au système, la seconde sera toujours complètement décentralisée.

À cette fin, nous avons adapté la solution Lehmann et Rabin au processus de minage en remplaçant des philosophes par des groupes de minage et en remplaçant les trois états des philosophes (faim, manger et penser) par trois états pour les groupes de minage (faim de miner, miner, repos pour refroidir les équipements et cesser de consommer de l’électricité).

La cueillette de deux baguettes (droite et gauche) est une condition requise pour lancer la course à l'exploitation du minage. Cette condition empêchera tout groupe de minage, quelle que soit sa capacité de hachage, de détourner le processus du minage et donnera à tout groupe de minage une chance équitable de gagner une récompense dans le système.

Compte tenu des résultats obtenus, nous avons conclu que la solution randomisée du problème du Dîner des philosophes, implémentée au sein d’un nœud arbitre empêcherait le monopole du processus de minage pour tout groupe de minage disposant d’une capacité de taux de hachage supérieure à 50%.

III.4 Article 4

Notre quatrième contribution, intitulée «Analyse du centre de gravité Bitcoin», répond à la quatrième question: «Qu'est-ce que le centre de gravité Bitcoin et que faut-il faire pour plus de sécurité?»

Nous avons procédé à une analyse des trois facteurs critiques pour Bitcoin système, à savoir: capacités critiques (CC), exigences critiques (CR) et vulnérabilités critiques (CV).

Nous avons obtenu le résultat suivant :

TABLEAU IV : ANALYSE DES FACTEURS CRITIQUES ET DU CENTRE DE GRAVITÉ (CDG) DE BITCOIN

<p>Centre de gravité (CDG)</p> <ul style="list-style-type: none"> • CDG stratégique de Bitcoin: <p>La confiance que l'on a gagnée parmi ses utilisateurs;</p> <ul style="list-style-type: none"> • CDG opérationnel de Bitcoin: <p>Le mécanisme de consensus (preuve du travail) qui permet au système de confirmer les transactions, de créer un nouveau bloc et de générer de nouveaux bitcoins.</p>	<p>Capacités critiques</p> <ul style="list-style-type: none"> • Créer de nouvelles unités de devise (bitcoins créés au cours du processus de minage) • Envoyer / recevoir des bitcoins (utilisation de Bitcoin comme système de paiement)
<p>Vulnérabilités critiques</p> <ul style="list-style-type: none"> • Vulnérabilité aux attaques de la majorité • Perte de clés privées accidentellement ou en cas de décès du propriétaire • Attaques par déni de service 	<p>Exigences critiques</p> <ul style="list-style-type: none"> • Réseau pair-à pair de Bitcoin • Applications de portefeuille • Procédé de minage

IV. Conclusion

Cette thèse a fourni un aperçu général sur la technologie Bitcoin. Elle a souligné certaines limitations de Bitcoin, telles que le débit maximal de sept transactions par seconde, la latence de confirmation et le gaspillage d'énergie pour les mineurs; des problèmes de sécurité, tels que la sécurité des transactions à zéro confirmation, le problème de fourchettes du Blockchain, la malléabilité de la transaction et l'attaque à 51%; et des applications telles que le stockage décentralisé, la gestion des identités et les contrats intelligents. En outre, elle a examiné les risques pertinents à la sécurité du Bitcoin en tant que crypto-monnaie et en tant que système de paiement. Pour cette fin, elle a identifié les principaux événements redoutés pour les actifs principaux, les scénarios de menaces pour les actifs secondaires prenant en charge Bitcoin et les risques estimés à l'aide de la méthode française d'évaluation des risques de la sécurité des systèmes d'information connue sous le nom d'«EBIOS». En outre, cette thèse a tenté de démontrer l'importance de la fonction de difficulté dans la sécurité de Bitcoin et la manière dont elle est ajustée dynamiquement pour éviter le détournement du Blockchain. En outre, les chercheurs ont effectué une analyse du centre de gravité de Bitcoin en analysant ses facteurs critiques et ont mis en évidence certaines stratégies pouvant être utilisées par des personnes malveillantes pour perturber l'écosystème Bitcoin. En outre, le document de recherche a également suggéré un moyen de contourner l'attaque de 51% en utilisant une solution du problème du dîner des philosophes qui devrait encore être développée pour aboutir à une solution décentralisée.

Un ensemble de mesures de sécurité ont été suggérées à la communauté Bitcoin et aux utilisateurs de Bitcoin afin d'atténuer les risques estimés. Leur mise en œuvre implique l'engagement des parties prenantes à tous les niveaux. Les utilisateurs de Bitcoin doivent être conscients des principes de base pour la sécurité de leurs portefeuilles et en particulier pour la sécurité de leurs clés privées, qui représentent un atout essentiel pour la sécurité de leurs fonds. Les nœuds Bitcoin doivent appliquer des contrôles de sécurité préventifs, tels que la mise à jour fréquente de leurs logiciels anti-virus, renforçant ainsi la sécurité du réseau pair-à-pair. Les développeurs de Bitcoins devraient continuer de corriger les failles et les bogues du système tout en accordant la priorité à la sécurité. En outre, les parties prenantes devraient investir davantage dans le domaine du minage afin d'accroître la sécurité du Blockchain et du mécanisme de consensus, qui est la pierre angulaire de tout le système. Ce dernier protégerait le système Bitcoin de la cupidité de certains mineurs qui pourraient s'entendre pour obtenir une majorité malveillante susceptible de nuire à la sécurité du Blockchain.

Enfin, nous pensons que Bitcoin demeurerait la crypto-monnaie la plus sécurisée du marché en raison de la preuve de travail, des fonctions de hachage, de la signature numérique à courbe elliptique et de l'implication de ses développeurs et de leurs réactions rapides aux défauts et aux erreurs. Cependant, une évaluation fréquente des risques de la technologie permettrait de mettre en évidence les principaux problèmes de sécurité et de prendre des mesures préventives adéquates de manière proactive.

Les applications fascinantes de la technologie Blockchain, en particulier pour le stockage décentralisé afin de résoudre les problèmes de falsification et de fraude, doivent faire l'objet de nouvelles recherches. Nous suggérons aux chercheurs d'étudier les possibilités de création d'applications distribuées qui stockent des diplômes universitaires et des titres immobiliers à l'aide de la technologie Blockchain, notamment Ethereum qui dispose d'un langage de programmation complet et adapté à ce genre de travaux

TABLE OF CONTENTS

TABLE OF CONTENTS

<i>Résumé</i>	i
<i>Abstract</i>	ii
ACRONYMS	v
ILLUSTRATIONS	vi
Chapter I Introduction	9
Chapter II Literature review	11
Chapter III Bitcoin technology, risk analysis methods, security issues and Bitcoin applications	15
III.1 Blockchain cryptographic constructs.....	16
III.1.1 Hash functions.....	16
III.1.2 Hash pointers.....	17
III.1.3 ECDSA.....	18
III.2 How Bitcoin works	19
III.2.1 Bitcoin keys and addresses	20
III.2.2 Key compression.....	25
III.2.3 Other types of keys and addresses	26
III.2.3 Bitcoin wallets.....	27
III.2.4 Bitcoin transaction	36
III.2.5 Bitcoin network.....	41
III.2.6 Bitcoin’s Blockchain.....	42
III.2.7 Bitcoin’s block	44
III.2.8 Proof of work and mining process	45
III.3 Risk analysis	51
III.3.1 Risk assessment methods for information security.....	51
III.3.2 Some risk management methods.....	51
III.3.3 EBIOS risk assessment method	52
III.4 Bitcoin limitations and security issues	54
III.4.1 Bitcoin limitations.....	54
III.4.2 Bitcoin security issues.....	54
III.5. Bitcoin applications	58
III.5.1 Decentralized storage.....	58

III.5.2 Decentralized identity management.....	58
III.5.3 Bitcoin smart contracts.....	59
Chapter IV Results	60
IV.1 Answer to the first research question	61
IV.1.1 Context of the study	61
IV.1.2 Study of the feared events.....	68
IV.1.3. Study of threat scenarios.....	69
IV.1.4 Study of risks	70
IV.1.5. Security controls	75
IV.2 Answer to the second research question.....	78
IV.3 Answer to the third research question	81
IV.3.1 Bitcoin majority attack.....	81
IV.3.2 Dining philosophers’ problem	81
IV.3.3. Contribution	83
IV.4 Answer to the fourth research question	89
IV.4.1. Bitcoin embedded security features	89
IV.4.2 Center of gravity analysis	91
IV.4.3 Bitcoin disruption strategies	92
V. Conclusion.....	94
APPENDIXES.....	Erreur ! Signet non défini.
Appendix I Table I Base58 symbol chart used in Bitcoin [].	98
Appendix II Fig.27 Execution of validation script	99
Appendix III Fig.28 ECDSA signing function used in Bitcoin.....	101
Appendix IV Fig.29 Verifying function of the ECDSA.....	102
Appendix V Fig.31 Various types of nodes in an extended Bitcoin network.	103
Appendix VI Risk analysis of Bitcoin security using EBIOS method	104
I. Introduction	104
II. Risk Assessment Methods for Information Security	105
III. Risk Study	106
III.2 Study Of The Feared Events	109
III.3. Study Of Threat Scenarios.....	109
III.4 Study Of Risks.....	110
IV.5. Security Controls	112

V. Conclusion.....	113
Appendix VII Bitcoin difficulty, a security feature	114
Appendix VIII Using the randomized solution of the dining philosophers’ problem to prevent the Bitcoin majority attack.....	118
Nomenclature	118
I. Introduction	118
II. Bitcoin Difficulty.....	120
III. Bitcoin Mining Pools.....	121
IV. Dining Philosophers Problem	121
V. Contribution.....	122
VI. CONCLUSION.....	125
Appendix IX Bitcoin embedded security items review and center of gravity analysis	126
I. INTRODUCTION.....	127
II. BITCOIN EMBEDDED SECURITY FEATURES	127
A. <i>Bitcoin Keys and Addresses Security</i>	127
B. <i>Bitcoin Wallet Security</i>	128
C. <i>Bitcoin Scripting Language</i>	129
D. <i>Bitcoin Transaction security</i>	129
E. <i>Bitcoin Blocks Security</i>	129
F. <i>Bitcoin Blockchain Security</i>	129
G. <i>Mining process Security</i>	129
III. CENTER OF GRAVITY ANALYSIS	129
A. <i>Definitions</i>	129
B. <i>Critical factors analysis</i>	129
C. <i>Bitcoin disruption strategies</i>	130
IV. CONCLUSION	130
REFERENCES.....	132

ACRONYMS

AES: Advanced encryption standard.

AKA: also known as.

ANSSI: French national agency for information systems security (Agence nationale de la sécurité des systèmes d'information).

BIP: Bitcoin improvement protocol.

BTC: Bitcoin.

CLUSIF: French information security club (Club de la sécurité de l'information français).

CLUSIQ: Quebec information security club.

CPOW: Computational power.

DDOS: Distributed denial of service.

DDP: Dining philosophers' problem.

EBIOS: Expression of needs and identification of security objectives (Expression des Besoins et Identification des Objectifs de Sécurité).

ECDSA: Elliptic curve digital signature algorithm.

ISO: International organization for standardization.

ISS: Information systems security.

NIST: National institute of standards and technology.

P2P: Peer-to-peer.

P2PKH: Pay to public key hash.

P2SH: Pay to script hash.

PM: Primary asset.

POW: Proof-of-work.

RIPEMD-160: RACE integrity primitives evaluation message digest 160 bits.

SA: Supporting asset.

SHA-256: Secure hash algorithm 256 bits.

TX: Transaction.

UTXO: Unspent transaction output.

WIF: Wallet import format.

ILLUSTRATIONS

Figure 1: Hash functions transform any data of any size to a fixed hash value.

Figure 2: Hash pointer.

Figure 3: Blockchain of chained blocks using hash pointers.

Figure 4: Bitcoin peer-to-peer network and its nodes.

Figure 5: A private key in hexadecimal - 64 characters in the range 0-9 or A-F.

Figure 6: A private key displayed in WIF format and QR-code format.

Figure 7: secp256k1's elliptic curve $y^2 = x^3 + 7$ over the real numbers.

Figure 8: Graph generated with a prime number $p=31$.

Figure 9: The process of private key, public key, and Bitcoin address derivation.

Figure 10: Base58Check encoding process.

Figure 11: Uncompressed and compressed public key.

Figure 12: Private key displayed in different formats.

Figure 13: Private and public key generation process.

Figure 14: BIP-38-private key encryption.

Figure 15: Some forms of Bitcoin wallets.

Figure 16: HD wallet- A tree of keys originating from one seed.

Figure 17: The process of generating a mnemonic sentence from a random number of 128 bits.

Figure 18: The process of generating a strong seed from a mnemonic sentence.

Figure 19: The process of creating the master keys and master chain code.

Figure 20: Child private key generation process.

Figure 21: Derivation of non-hardened child public keys and child chain code from an extended public key.

Figure 22: Hardened derivation of a child key.

Figure 23: Paper wallet - BIP38-encrypted private key and Bitcoin address.

Figure 24: An example of brain wallet.

Figure 25: An example illustrating the structure of a real Bitcoin transaction.

Figure 26: The concatenation of an unlocking script and a locking script.

Figure 27: Execution of validation script.

Figure 28: ECDSA signing function used in Bitcoin.

Figure 29: Verifying function of the ECDSA.

Figure 30: The four types of functionalities played by Bitcoin nodes.

Figure 31: Various types of nodes in an extended Bitcoin network.

Figure 32: Chain of blocks forming the Blockchain.

Figure 33: Bitcoin fork-the longest chain is the valid Blockchain.

Figure 34: The Blockchain increasing size.

Figure 35: Block structure.

Figure 36: The process of creating a Merkle tree of a block of transactions.

Figure 37: Bits value field of the block #0.

Figure 38: Difficulty calculation of block #495223.

Figure 39: Block #495223 information.

Figure 40: ASIC machine used in mining.

Figure 41: Mining warehouse made of ASIC machines.

Figure 42: Hash-rate distribution amongst the largest mining pools.

Figure 43: Phases of EBIOS method.

Figure 44: Decentralized storage in Bitcoin.

Figure 45: Perimeter of the study.

Figure 46: Risk assessment illustration

Figure 47: Difficulty calculation of block #495223.

Figure 48: Block #495223 Information.

Figure. 49: Eight sitting dining philosophers.

Figure.50: A basic solution of the DPP.

Figure 51:.A deadlock in the DPP.

Figure.52: A solution to the deadlock problem in the DPP.

Figure 53: Starvation issue in the DPP.

Figure 54: Arbitrating node in Bitcoin.

Figure 55: State diagram of the process.

Figure 56: Randomized solution of the DPP adapted to Bitcoin mining process.

Figure 57: Model checking.

Figure. 58: Miners' different states (0 to 11) for 100 iterations.

Figure 59: Results of the model checking properties.

Figure. 60: Excerpt of code and results used for testing.

Figure 61: Randomized order of mining for the 19 mining pools different for each attempt.

Figure.62: Secp256k1 defined elliptic curve.

Figure 43: Threat sources.

INTRODUCTION

INTRODUCTION

Introduction

Blockchain is a distributed ledger that stores transactions in remote nodes, connected to each other through a peer-to-peer network. In other words, it can be seen as a database file that is shared between different connected computers around the world. A file that records all the transactions that have ever happened between the users. This new technology has removed the need for a trusted third-party to process transactions, thus revolutionizing many aspects of our daily life, especially business and finance.

This technology has set the stage for what is widely known as crypto-currencies and payment systems. *Satoshi NAKAMOTO* was the first to suggest a decentralized payment system and a virtual crypto-currency, known as Bitcoin, based on the Blockchain technology. In his white paper, *Bitcoin: A Peer-to-Peer Electronic Cash System*, *NAKAMOTO* proposed in 2008 a solution to the double-spending problem through the use of a peer-to-peer network and a hash-based proof of work [1].

Bitcoin was forged over cutting-edge cryptographic constructs, such as hash functions and digital signatures. *NAKAMOTO* made the first implementation of this new technology to demonstrate that the concept is feasible and workable. This made an important cornerstone for the growth of the Blockchain technology. In addition, the open-source nature of the implementation draw the interest of volunteering developers who contributed efficiently to this effort.

The core premise of Bitcoin was to allow two or more users to send and receive anonymously transactions without relying on a third-party to process them. Payment and money exchange are made in a decentralized way. Transactions are disseminated and validated collectively in a peer-to-peer network of participating nodes, which makes the Bitcoin transaction fees lower than any financial institution around the world.

The original design promised to provide two appealing features ease of use and anonymity. Bitcoin users could exchange bitcoins relying on pseudonyms, known as Bitcoin addresses, without revealing their identities. Bitcoin users participate collaboratively in confirming the transactions. Transactions are broadcasted to the whole peer-to-peer network and validated by all users. Then, they are confirmed in blocks by peers or miners who compute to solve a computational puzzle. The successful peers are automatically rewarded with new bitcoins. This serves as an incentive for miners to stay honest and continue confirming the transactions. Also, the computational puzzle serves as a decentralized time-stamping service in the network, and as a decentralized consensus over the transactions and blocks, which deters many attacks on the system.

Notwithstanding its embedded security features, many researchers have reported numerous types of attacks against Bitcoin system, such as double-spending, eclipse attacks, selfish mining, and lack of privacy. These issues are examined in detail in this paper. In addition, disagreements among Bitcoin community of developers on the outlook of this technology, especially on whether or not to extend the block' size to sustain its ever-growing development, brought about the split of Bitcoin in three different forks which are the Bitcoin Core, The Bitcoin Classic, and The Bitcoin XT (also known as Bitcoin Cash).

Despite the aforementioned issues, Bitcoin gained a wider community of users and has sustained 10 years of operation as the most prominent cryptocurrency. This sustainability is more likely due to the swift interaction between Bitcoin developers and the research community in tackling flaws and issues to save the system from a wide range of attacks. At the time of writing, Bitcoin holds the

largest market share with more than 114 billion US dollars of market capitalization and an exchange price of around 6600 dollars [2].

Bitcoin's success set the stage for other alternative currencies, aka altcoins, which most of them are just clones of Bitcoin with some slight alterations. There are currently 2042 crypto-currencies [3]. Crypto-currencies differ from each other based on the following main properties: the circulating coin supply, the maximum coin supply, the hash function, the block size, and the block generation time.

Some of the most prominent Blockchain proposals, are but not limited to, Namecoin, Litecoin, and Ethereum. Namecoin is the first clone of Bitcoin that stores web addresses that end with “.bit” and works as a decentralized domain name service that provides some resiliency against censorship. Litecoin is a Bitcoin clone that generates blocks in every 2.5 minutes and relies on a script-based proof-of-work as a consensus mechanism. Ethereum is a POW-based Blockchain that relies on a complete programming language besides the implementation of smart contracts, which are applications that run on the Blockchain. The scope of this research thesis is limited to the study of Bitcoin.

In this research paper, we examine the most relevant Blockchain limitations, Security issues and applications related to Bitcoin technology. More importantly, we are endeavoring to answer the following key questions:

- What are the major risks related to using Bitcoin as a crypto-currency and as a payment system?
- What is the relation Between the Bits, the Target, and the Difficulty; and How the Bitcoin difficulty contributes to the security of the system?
- How can we deter the majority attacks or the selfish mining?
- What are the Bitcoin embedded security items and what is the Bitcoin center of gravity; and what should be done to disrupt or secure the system?

The aim of this paper is to provide some insights about Bitcoin limitations, security risks, and applications. It is organized as follows. Section 2 reviews some research papers related to the Bitcoin security challenges, and security risks. Section 3 introduces Bitcoin technology, examines its security and outlines its applications. Section 4 presents the results of our research journey and provides answers to the thesis questions. Section 5 provides a summary of the whole work and an outlook for further research on the Blockchain applications.

Chapter I

Literature review

So many papers related to the Blockchain technology are available online for the research community. We selected only papers that are relevant to Bitcoin security, and its security risks. In this section, we review these papers and outline their relevancy to our research question. In addition, this section will provide to the reader more theoretical background about this subject-area.

Satoshi NAKAMOTO, the Bitcoin white paper author, mentioned that Bitcoin transactions security depends on who detains the majority of the network computational power. He pointed out that the honest miners will always prevail and produce the longest chain since the system provides an incentive. He calculated and showed that the probability of a slower attacker (detaining less than 50% of the hashing power), wanting to change previous blocks, to catch up with the system drops potentially when six blocks or more are mined [4]. Despite this underlying assumption, Bitcoin is still vulnerable to dishonest miners, detaining more than 50% of the network hashing power, who can theoretically mess with Bitcoin security by making double-spending transactions, preventing transactions confirmations, monopolize the mining process and preventing other honest miners from creating new blocks, etc.

Ittay Eyal & Emin Sirer in their paper, “Majority is not enough: Bitcoin Mining is Vulnerable”, suggest that selfish miners who are detaining more than 33% of the network hashing power can still acquire an important part in the mining process. They mentioned that a selfish mining strategy consists of a miner not announcing his mined blocks to the network in order to increase their revenue and letting other miners wasting their time and computational power. They suggested a countermeasure to prevent the aforementioned strategy by urging miners to disseminate all the received blocks and choose randomly one block to mine on it in case of two competing blocks [5].

Arthur Gervais, Hubert Ritzdorf, Ghassan O. Karame, and Srdjan Capkun in their paper, “Tampering with the Delivery of Blocks and Transactions in Bitcoin”, declared that an attacker even with constrained-resources could find a way around the “*Eyal & Sirer*” security measure by exploiting the Bitcoin object request management system, which would prevent blocks delivery for around 20 minutes. They demonstrated feasibility and easy realization of their attacks in current Bitcoin client through analysis and implementation of some hosts. They showed that the adversary can easily mount Denial-of-Service attacks, considerably increasing his mining advantage in the network or double-spend transactions in spite of the current countermeasures adopted by Bitcoin. Their contribution consists of a modification of the block request management system in Bitcoin in order to detect any misbehavior in the delivery of blocks and harden the security of the network without deteriorating the scalability of Bitcoin [6].

Ayelet Sapirshstein, Yonatan Sompolinsky, and Aviv Zohar in their paper: “Optimal selfish mining strategies in Bitcoin”, defined a lower threshold of CPOW (lower than the one defined by *Eyal & Sirer*) at which selfish miners could be successful. They cited that attackers with strictly less than 25% of the computational resources can still gain from selfish mining, unlike what *Eyal & Sirer* conjectured. In addition, they demonstrated how any attacker for which selfish mining is profitable can execute double spending attacks bearing no costs, unlike what the security analysis of *Satoshi NAKAMOTO* has guessed [7].

Nicolas T. Courtois, Lear Bahack in their paper: “On Subversive Miner Strategies and Block Withholding Attack in Bitcoin Digital Currency”, studied in details several recent attacks in which dishonest miners obtain a higher reward than the irrelative contribution to the network. They also

revised the concept of block withholding attacks discussed by [5, 6, 7] and proposed a new concrete and practical block withholding attack which maximize the advantage gained by rogue miners [8]

Ittay Eyal, in his paper: “The Miner’s Dilemma”, explored a block withholding attack among Bitcoin open mining pools when attackers try to increase their revenue. In addition, he alluded that mining pools can increase their revenue by making some of their miners infiltrate other mining pools and withhold blocks. Moreover, He mentioned that attacked pools cannot detect which of the miner is attacking it, and thus their revenue density could reduce and eventually miners would leave this pool or join a closed mined pool limited to trusted participants. This would lead to smaller pools, and so ultimately to a better environment for Bitcoin as a whole since large pools hinder the distributed nature of the system. Also, he mentioned that algorithms that distinguish partial from full proofs of work do exist but their use would not be in the interest of the community since they would reduce or remove block withholding and therefore encourage large pools [9].

Zhenzhen Jiao and Rui Tian and Dezhong Shang and Hui Ding, in their paper: “Bicomp: A Bilayer Scalable Nakamoto Consensus Protocol”, discussed how Bicomp can resist selfish mining. Their approach is based on high security and pure decentralized Nakamoto consensus, and with a significant improvement on scalability. In Bicomp, two kinds of blocks are generated, i.e., micro-blocks for concurrent transaction packaging in network, and macro-blocks for leadership competition and chain formation. A leader is elected at beginning of each round by using a macro-block header from proof-of-work. An elected leader then receives and packages multiple micro-blocks mined by different nodes into one macro-block during its tenure, which results in a bilayer block structure. Such design limits a leader’s power and encourages as many nodes as possible to participate in the process of packaging transactions, which promotes the sharing nature of the system and resists to selfish mining [10].

In another work entitled, “Resisting Selfish Mining Attacks in the Bicomp”, Rui Tian and Wei Gong mentioned that the selfish mining strategy can compromise a Nakamoto consensus system with less than 25% mining power of the whole system. They have analyzed in detail the selfish mining resistance of the Bicomp protocol. Through modeling the system as a state machine, and analyzing different mining activities that lead to state transition, they concluded that the Bicomp can adjust its resistance towards selfish mining attack by varying macroblock POW difficulty and tenure length parameters. They also presented a modification to the Bicomp protocol without substantially modifying the operation mechanism of the system [11].

Jaewon Bae and Hyuk Lim, in their article entitled: “Random Mining Group Selection to Prevent 51% Attacks on Bitcoin”, mentioned that an attacker node whose hash power is greater than half of the total hash power, that node can perform a double-spending attack, i.e., a 51% or majority attack. They proposed an approach to reduce the probability of a successful double-spending attack on Bitcoin. The proposed approach divides miners into groups and gives mining opportunity to a randomly selected group. Their analysis showed that if the number of groups is greater than or equal to two, the probability that the attacker will find the next block is less than 50%. They concluded that this approach can reduce likelihood of a majority attack and can reduce the computing power costs of block mining [12].

C. Decker and R. Wattenhofer in their paper entitled, “Bitcoin Transaction Malleability and MtGox”, Studied transaction malleability in Bitcoin. They examined a real study case of *MtGox* which claimed to have lost 850,000 bitcoins due to malleability attacks. They observed through analysis that only a total of 302,000 bitcoins were being involved in malleability attacks. They claimed too that

78.64% of these attacks were ineffective before concluding that barely 386 bitcoins could have been stolen using malleability attacks from *MtGox* or from other businesses. They stated that transaction malleability is a real problem and should be addressed in any Bitcoin client implementation [13].

M.Andrychowicz, S. Dziembowski, D.Malinowski, and L. Mazurek in their paper entitled, “On the Malleability of Bitcoin Transactions”, performed practical experiments on Bitcoin that showed high probability of Bitcoin transactions malleability. They analyzed the behavior of the popular Bitcoin wallets in the situation when their transactions are being malleable. They concluded that most of Bitcoin wallets of that time were to some extent not able to handle this situation correctly. For this purpose, they suggested a deposit protocol with a timed commitment scheme to create a malleability-resilient “refund” transaction which does not require any modification of the Bitcoin protocol [14].

Mariam Kiran and Mike Stannett, in their paper “Bitcoin Risk Analysis”, studied different risk areas related to Bitcoin such as social, legal, economic, technological, and security risks. In security risks, they pointed out three high-level risks pertaining to Bitcoin security, which are man-in-the-middle attacks, loss of keys, and the subversive miner strategies [15].

Jerry Brito and Peter Van Valkenburgh, in their paper “Bitcoin: Risk Factors For Insurance”, mentioned six global threats that could harm the functioning of the Bitcoin. These threats are: flawed key generation, transaction malleability, 51% attacks, Sybil attacks, DDOS attacks, and consensus or fork risks [16].

Chapter II

Bitcoin technology, risk analysis methods, security issues and Bitcoin applications

This section intends to provide some background information that is relevant to the Blockchain technology, mainly Bitcoin so that the reader would gain more insights about how the technology works, its security implications and its main applications. The first sub-section outlines some required cryptographic constructs for better understanding Blockchain technology. For this purpose we examine hash functions, hash pointers, and the Elliptic Curve Digital Signature Algorithm (ECDSA). The second sub-section provides some details on how Bitcoin works. The third sub-section scrutinizes Bitcoin limitations, Security issues, and applications.

Before diving into Blockchain architecture, we will delve into these cryptographic tools that are implemented in the system.

II.1 Blockchain cryptographic constructs

II.1.1 Hash functions

Hash functions, as shown in Fig.1, are used to map any data of any size to a fixed size output, which is known as a digest, a hash code or a hash value. They hold the characteristic of being **one-way** functions and thus it is computationally infeasible to build the input data that produced a given hash value. In other words given a hash code “Y”, it is almost impossible to find a value X such that

$$F(X) = Y$$

This property is also known as “hiding property”. We can try a brute-force attack on the hash function, but this will take millions years to find the value of “X”, which is not interesting for the attacker. This feature made hash functions very useful in computer science, especially as strong way to check the integrity of the data exchanged in the network.

Another interesting property that hash functions promise is the “collision resistance”, which makes it computationally infeasible to find two different inputs that map to the same hash value. Note that collisions do exist since hash functions transform any data of any size to a digest of fixed-size. Despite this fact, collisions are still almost impossible to find giving the current computational power. If we try 2^{130} randomly chosen inputs, we have a probability of 99.8% that two of them will collide [17]. 2^{130} is an astronomical number which takes billions of years to calculate. No hash function is proven to be collision free, but still collisions are almost infeasible to find till now. This second property is very important, since Bitcoin’s transactions identifiers (TXIDs) are the hashes of signed transactions. Hashing is also used in the Bitcoin’s proof-of-work mechanism.

Also, hash functions provide an important property that is being used in the Bitcoin POW-based consensus. This property is “puzzle-friendly”. A hash function H is said to be puzzle-friendly if for every possible n-bit output value y, if k is chosen from a distribution with high min-entropy, then it is infeasible to find x such that $H(k | x)=y$ in time significantly less than 2^n .

Bitcoin uses SHA-256 and RIPEMD160 hash functions, while Ethereum relies on SHA-3 for 256-bit and 512-bit outputs, which provides more security. SHA 256 was designed by the US National security agency (NSA). It stands for secure hash algorithm and produces a 256-bit hash value.

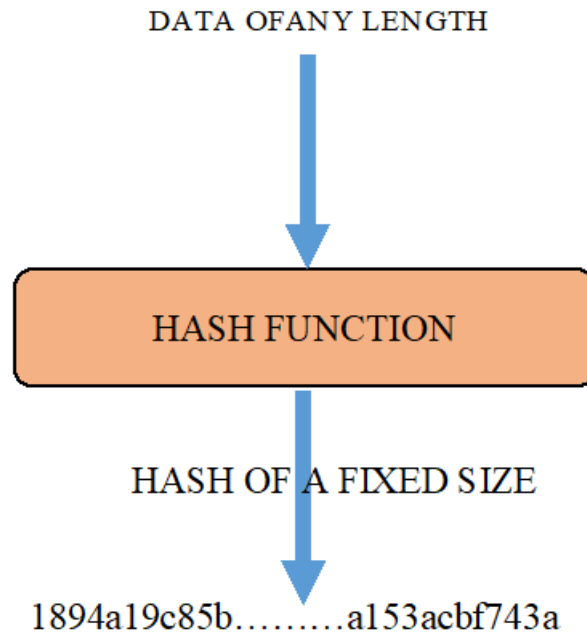


Fig.1.Hash functions transform any data of any size to a fixed hash value.

II.1.2 Hash pointers

A Hash pointer is data structure that points where some information is stored and holds also a hash of the information, which helps detect and prevent tampering with the data (See Fig.2). This data structure is used to build the Blockchain and the Merkle Tree.

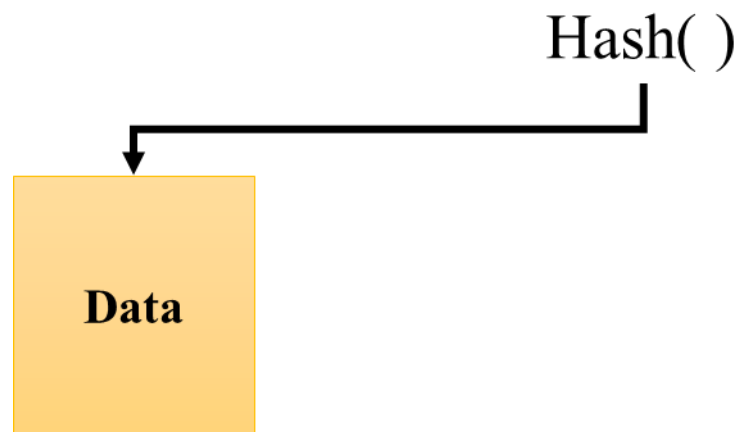


Fig.2. Hash pointer.

III.1.2.1 Blockchain structure

Blockchain is data structure in the form of a back-chained list of blocks using hash pointers. Each block consists of data and a hash of the previous block's data. Hash pointers provides a way to detect any change in the blocks data. If an attacker tampers with the data of a block, this will make the stored hash pointer of the next block inconsistent with the actual one, then they have to change all the hash pointers in the following blocks. Despite all this, the head pointer will not match with the stored one and thus the tampering will be detected. (See Fig.3)

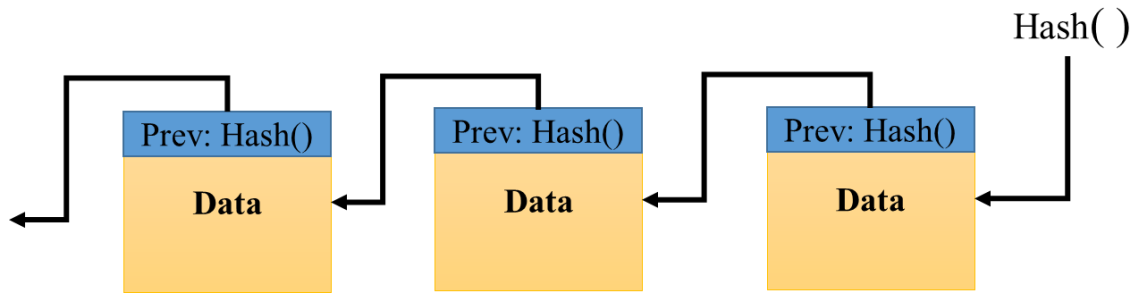


Fig.3. Blockchain of chained clocks using hash pointers

In Bitcoin, the Blockchain is shared among peers in a distributed way, which makes it so hard to tamper with. If someone succeeds in tampering with a copy of the Blockchain, he will have to tamper with all the other copies of the Blockchain available in the network, which is almost impossible.

III.1.2.2 Merkle tree

A Merkle tree is a hash-based data structure organized in the form of a tree, in which each leaf holds a hash of a block of data, and each non-leaf node is a hash of its children [18]. It is used in distributed systems for data integrity verification. Bitcoin's block uses the Merkle tree to check the integrity of the transactions. More detail about the use of Merkle Tree in Bitcoin is provided later in this chapter.

II.1.3 ECDSA

Elliptic Curve Digital Signature Algorithm (ECDSA) is a cryptographic algorithm used by Bitcoin to ensure that funds can only be spent by their rightful owners [19]. It consists of three main concepts, which are private keys, public keys, and signatures. All these elements are explained later in details in this chapter.

II.2 How Bitcoin works

Bitcoin is totally a virtual currency that stores values in the form of transactions. Users hold digital wallets and use keys to unlock their transactions in order to spend or send its value to another participant. These keys serve also as a proof of ownership.

Bitcoin is a decentralized system, which means that there is no central bank or government agency that regulates its functioning. New bitcoins are created through the mining process. Miners with huge computational power compete to solve a mathematical puzzle so they can create a new block of transactions and get a new brand of bitcoins as a reward. A solution to this puzzle is found every 10 minutes on average.

The difficulty of this puzzle is set dynamically so a solution can be found within 10 minutes. Also the currency issuance, which is a reward for successful miners, is halved every 210,000 blocks created (in every four years on average), which limits the amount of bitcoins to 21 million bitcoins in total. This amount will be reached by the year of 2140. This rule makes Bitcoin a deflationary currency, but never inflationary. It helps also increase the value of the currency. Note that one bitcoin (BTC) can be subdivided into 100,000,000 satoshis.

Bitcoin can be considered as the best form of money for the internet since it allows a quick, secure, and borderless transfer of money. It uses a distributed, peer-to-peer network to disseminate the transactions among the participants. It relies on an open source software written in C++ language. In the following sections, we delve into the major technical details that form Bitcoin so the reader can get more insights about this technology.

Bitcoin is different from a traditional banking and payment system. It relies on decentralized trust in which participants are equal and play a major role. This trust is achieved through a mechanism of distributed consensus. Before we delve into the technology used in Bitcoin system, we need to provide a general overview of what it consists of (see Fig.4). Bitcoin is made of users interacting with the system through wallets containing keys. These wallets send and receive transactions that are disseminated to other nodes over a peer-to-peer network. This network is made of nodes that play different roles. Transactions are collected and validated by Miners who compete against each other to solve a mathematical puzzle, known as the proof-of-work. Once a miner succeed in finding the proof-of-work, he is able to create a new block of transactions. The newly created block is propagated through the network to almost all the active nodes. It is then added to a public ledger called the Blockchain. The Blockchain is made of a chain of blocks in which a block references its previous one, and so on, all the way back to the first block, known as the genesis block.

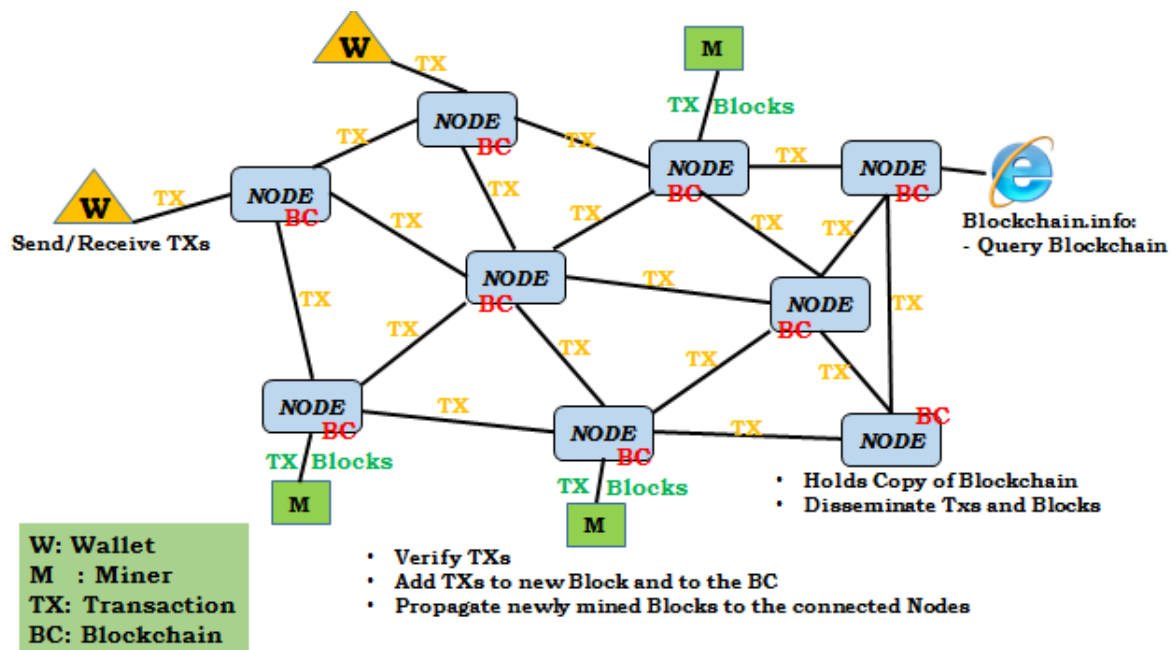


Fig.4 Bitcoin peer-to-peer network and its nodes.

In Bitcoin, transactions are sent to users using their addresses, which are made of alphanumeric string of characters. Addresses are derived from the private and public keys. Bitcoin wallets allow users to create and manage as many addresses as they want. This help increase the user’s anonymity in the system. For this purpose, Bitcoin users are strongly urged to use an address only once.

Bitcoin wallets create and store digital keys, addresses, and digital signatures, which serve as a proof-of-ownership. These keys consist of a pair of private key and public key. While the public key can be disclosed, the private key must remain secret in order to secure the funds. If a private key is lost or revealed, all the related funds are lost forever. In the following paragraphs, we will explain how Bitcoin keys and addresses are created, some different type of wallets, what transactions consist of and how they function, Bitcoin P2P network, Blockchain, Mining process and the consensus mechanism.

II.2.1 Bitcoin keys and addresses

While Bitcoin do not use cryptography to encrypt the transactions or the communications, it relies heavily on asymmetric cryptography to secure the ownership of funds through the use of digital keys, addresses, and digital signatures.

II.2.1.1 Private keys

The private key is used to create the signature, as a proof of ownership, which is also required to spend the bitcoins in a transaction. Private keys give full control over the funds, therefore they must be kept secret and also must be backed up in order to prevent any unintentional loss. The private key is generated using the entropy of operating system for both the deterministic and the non-deterministic wallets. The private key is a number of 256 bits that is picked randomly, between 1 and 2^{256} . The method used to generate this number must neither be predictable nor repeatable. It must use a source of sufficient entropy to generate a secure private key. Fig.5 shows a private key in hexadecimal (256 bits in hexadecimal is 32 bytes or 64 characters in the range 0-9 or A-F).

E9 87 3D 79 C6 D8 7D C0 FB 6A 57 78 63 33 89 F4 45 32 13 30 3D A6 1F 20 BD 67 FC 23 3A A3 32 62

Fig.5 A private key in hexadecimal - 64 characters in the range 0-9 or A-F [20].

Private keys are exported and imported in a standardized format, known as the Wallet Import Format (**WIF**), which is encoded in a Base58check format and uses a flag indicating whether or not the private key is related to a compressed public key. The Base58check format includes built-in error checking codes that automatically detect and correct errors. Fig.6 shows a private key, encoded in WIF format and a QR-code format, generated randomly from the bitaddress.org website.



Fig.6 a private key displayed in WIF format and QR-code format [21].

II.2.1.2 Public key

Bitcoin wallets use elliptic curve multiplication to produce public keys. They rely on an Elliptic Curve Discrete Logarithm formula, which is hard if not impossible to reverse giving the currently available computational power. This mathematical equation is defined as follows:

$$\text{PUBKEY} = \text{PRIVKEY} * G \quad (1)$$

Where G is the Generator point of the Koblitz Elliptic Curve, which is:

$$Y^2 = X^3 + 7 \quad (2)$$

Its recommended parameters are defined as secp256k1 by the Standards for Efficient Cryptography [22]. The public key security relies strongly on elliptic curve discrete logarithm problem. This means that despite knowing the public key and the generator G, it is still so hard or impossible to find the corresponding private key.

Unlike the most commonly-used elliptic curves that have a random structure, secp256k1 was built in a special non-random way which makes computation more efficient and reduces also the risk of any sort of backdoor into the curve[23]. The standard defines some parameters that are associated with a Koblitz curve F_p (See Fig.7), which are specified by the sextuple $T = (p,a,b,G,n,h)$ as follows [22]:

- **The Prime Number (very large)**
 $p = \text{FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFC2F}$ or
 $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1.$
- **The Elliptic Curve E**
 $y^2 = x^3 + ax + b$ over F_p is defined by: $a = 0; b = 7$

- **The Base Point G**

- In Compressed Form:

G = 02 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798

- In Uncompressed Form:

G = 04 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798 483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8

- **The Order n of G**

n = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141

The order n of a point G is the smallest integer different from zero that satisfy the following equation:

$$n * G = \text{point at infinity} \quad (3)$$

- **The Co-Factor**

The co-factor is the order of the entire group E divided by the order of the subgroup generated by the Base point G:

$h = (\text{cardinal of } E) / n = 1$. This means that G can produce the whole points of the Curve in the finite field E.

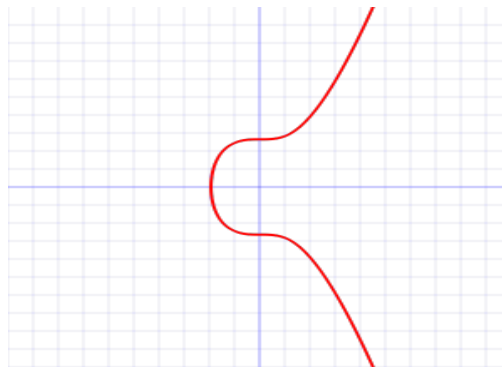


Fig.7. secp256k1's elliptic curve [$y^2 = x^3 + 7$] over the real numbers [24].

Since the elliptic curve is defined in a finite field of prime order, its curve should look like a pattern of scattered dots as depicted in Fig.8.

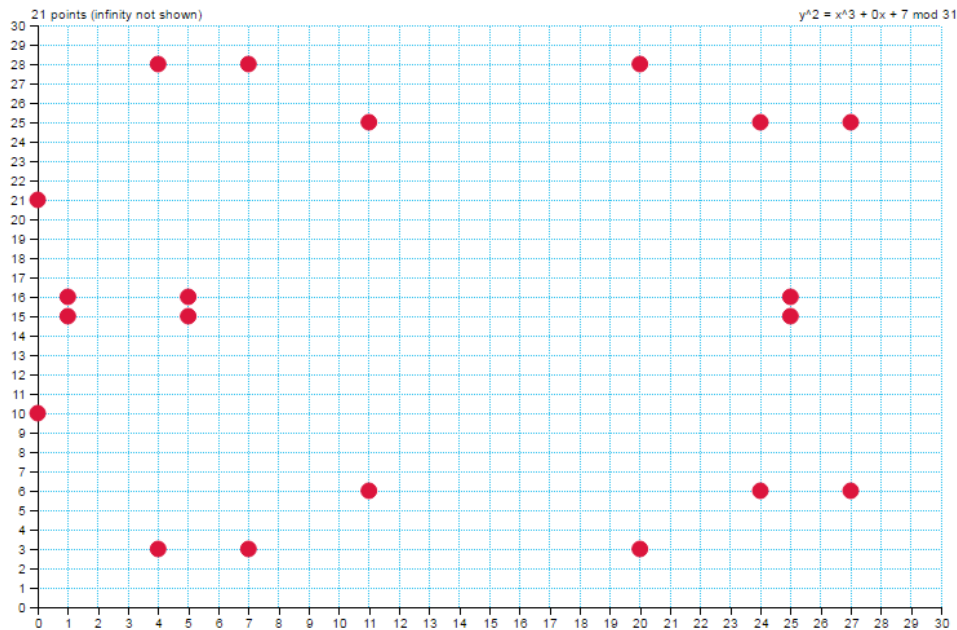


Fig.8. Graph generated with a prime number $p=31$ [25].

III.2.1.3 Bitcoin address

Bitcoin Addresses are derived from public keys using two hash-functions, SHA-256 (*Secure Hash Algorithm 256 bits*) and RIPEMD-160 (*RACE Integrity Primitives Evaluation Message Digest 160 bits*) as in the following formula:

$$\text{ADDRESS} = \text{RIPEMD-160}(\text{SHA-256}(\text{PUBKEY})) \quad (4)$$

This make it also irreversible since hash-functions are one-way functions. Fig.9 depicts the derivation of public keys and Bitcoin addresses from private keys.

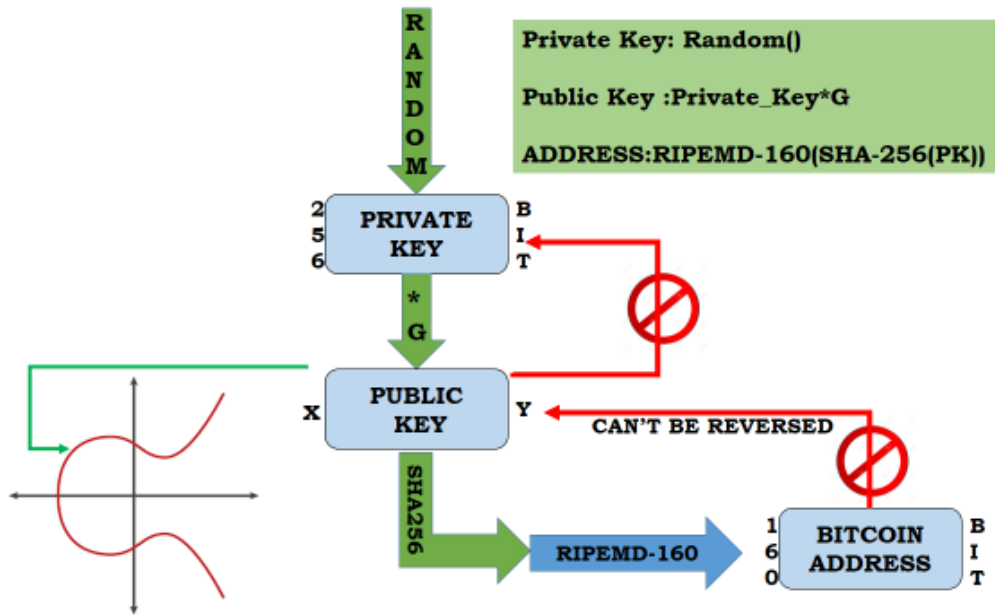


Fig.9. the process of private key, public key, and Bitcoin address derivation.

In Bitcoin, there is no account numbers, so users rely on addresses to send or receive bitcoins through transactions. The addresses serve as an identifier for users who are participating in the system. They help ensure the anonymity of the users who do not want to reveal their personal information when sending or receiving bitcoins in the network. Also and for the same purpose, users should use new addresses anytime they exchange bitcoins since transactions are made public and can be tracked to their originators.

Furthermore, Bitcoin uses the Base58Check encoding to prevent transcription errors and also to avoid sending bitcoins to an invalid address. This display formatting is explained in details in the following paragraphs.

III.2.1.4 Base58 encoding format, prefix version, and Base58check

III.2.1.4.1 Base58 encoding format

Similar to Base64, which is mostly used to add binary attachments to email, Base58 is a text-based binary encoding format developed specifically for Bitcoin and now is widely spread in other crypto-currencies. It uses a collection of 58 alphanumeric symbols consisting of easily distinguished uppercase and lowercase letters (00II are omitted). Table I (See Appendix I) illustrates the Base58

symbol chart used in Bitcoin. It links numbers from 0 to 57 with their corresponding number or letters in Base58 encoding format.

III.2.1.4.2 Prefix version

Before converting the data into a Base58Check format, we add a prefix called “the version byte”. This helps quickly identify the type of data. Table II shows some common version prefix types along with their resulting prefixes in Base58 format.

TABLE II. SOME COMMONLY USED VERSION PREFIX TYPES IN BITCOIN [26].

Type	Version prefix (hex)	Base58 result prefix
Bitcoin Address	0x00	1
Pay-to-Script-Hash Address	0x05	3
Bitcoin Testnet Address	0x6F	m or n
Private Key WIF	0x80	5, K, or L
BIP-38 Encrypted Private Key	0x0142	6P
BIP-32 Extended Public Key	0x0488B21E	xpub

III.2.1.4.3 Base58check

Base58check is a Base58 encoding format used to detect and prevent transcription errors in keys or addresses. It adds a checksum to the end of the data being encoded. This checksum consists of the first four bytes of double hash of the data being encoded using SHA-256 hash-function as shown in the following formula:

$$\text{CHECKSUM} = \text{SHA-256}(\text{SHA-256}(\text{PREFIX} + \text{DATA})) \quad (5)$$

Once the checksum is performed, we end-up with three items: the prefix version, the data, and the checksum. These three items are concatenated and passed through the Base58 encoding process. Fig.10 illustrates the whole process of Base58Check encoding.

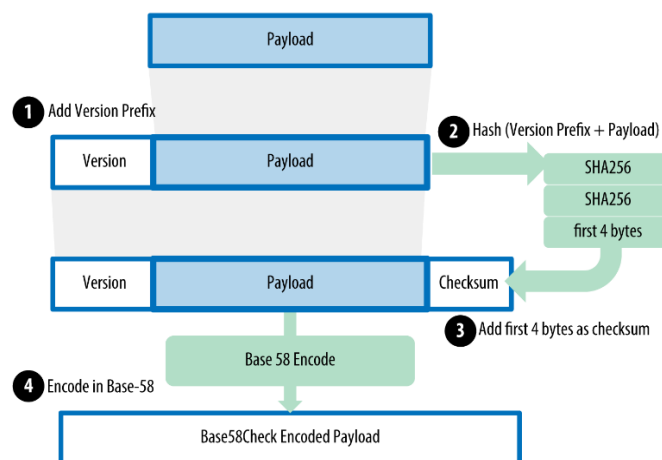


Fig.10 Base58Check encoding process [27].

II.2.2 Key compression

The main purpose of key compression is cut down the size of the transactions and save some space in the database that stores the Blockchain.

II.2.2.1 Public key compression

As we previously mentioned, the public key is generated from a private key using an elliptic curve discrete logarithm, which means that the public key is in the form of a point with two coordinates: X and Y. To produce an uncompressed Base58check public key, we need 520 bits of memory space representing the “Prefix+X+Y”. Since most of the transactions include the public key as a requirement for validating the user’s ownership, this raises the size of the transactions and therefore increase the size of the block and the Blockchain as well. To solve this issue, we rely on the elliptic curve equation described in the secp256k1 to calculate the Y for each X introduced. Therefore, we will use only the X in the public key and save 50% of the transaction size.

Since the curve is symmetrical over the x-axis, each given X of the curve will produce two solutions: one above the x-axis as a positive solution and the other one below the x-axis as a negative solution. So, if we have to omit the Y, we need to store its sign in the public key in order to easily find the right one. To overcome this issue, Bitcoin uses the even/odd technique with different prefixes. It uses “02” prefix if the Y is an even number, and “03” prefix if Y is an odd number. For the uncompressed public address, the prefix “04” is used. The compressed public key is 264 bits in size. Fig.11 shows the prefixes used in the uncompressed and compressed public key.

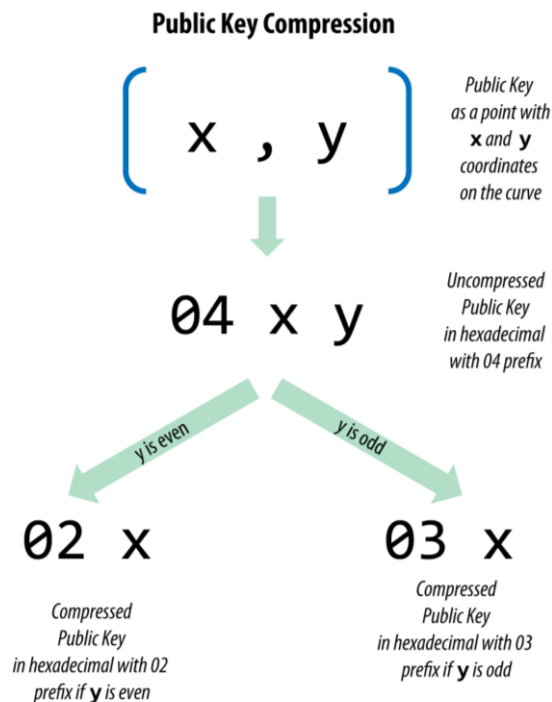


Fig.11 Uncompressed and compressed public key [28].

III.2.2.2 Private key compression

A compressed private key is a private key to which we add the suffix “01”. Ironically, a compressed private key is little longer than an uncompressed one. Private Key compression serves only to distinguish that a compressed public key was originated from a compressed private key in order to avoid the confusion to where funds should be sent whether to an address derived from a compressed

or an uncompressed key. WIF-compressed private keys start with a “K” or “L”, while the WIF-uncompressed (also referred to as WIF) private keys start with a prefix “5”. Fig.12 illustrates an example of private key displayed in different formats.

Format	Private Key
Hex	1E99423A4ED27608A15A2616A2B0E9E52CED330AC530EDCC32C8FFC6A526AEDD
WIF	5J3mBbAH58CpQ3Y5RNJpUKPE62SQ5tfcvU2JpbnkeyhfsYB1Jcn
WIF-compressed	KxFC1jmwwCoACiCAWZ3eXa96mBM6tb3TYzGmf6YwgdGWZgawvrtJ

Fig.12. Private key displayed in different formats [29].

The whole process of generating compressed or uncompressed private and public keys is illustrated in the following figure (Fig.13).

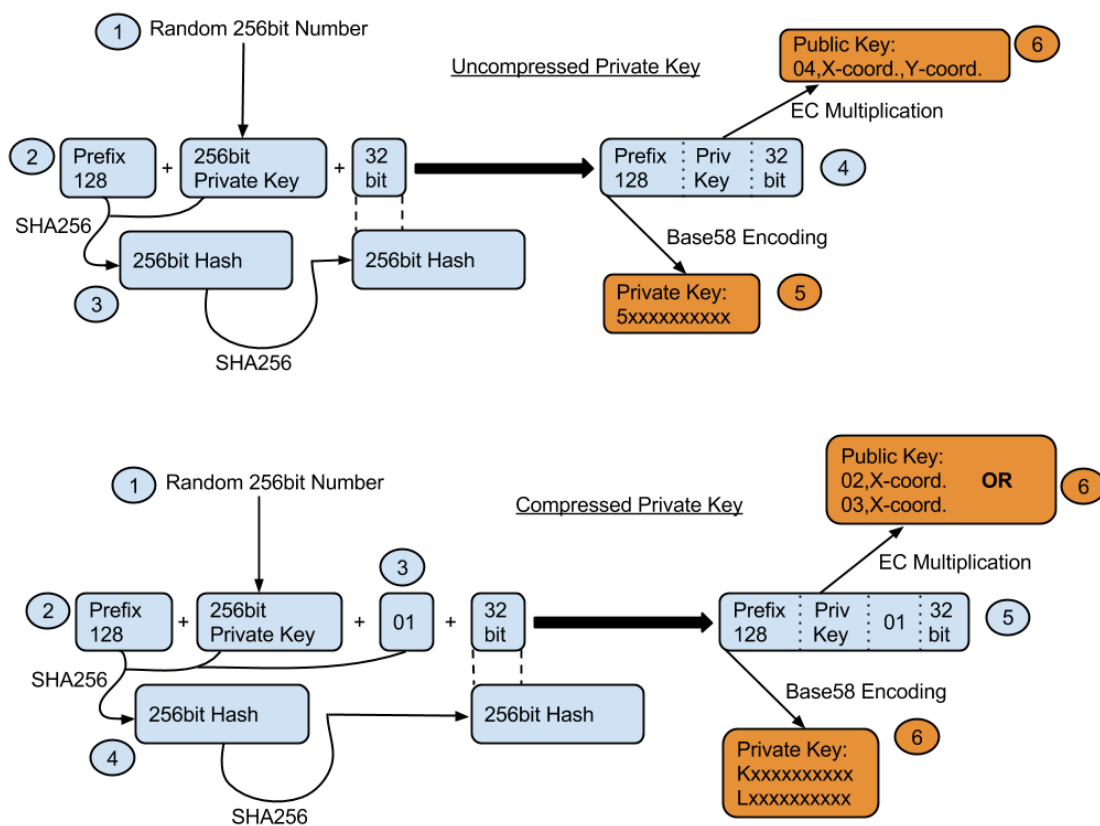


Fig.13. Private and public key generation process [30].

II.2.3 Other types of keys and addresses

In this section we will delve into some advanced types of keys and addresses that are used in Bitcoin such as encrypted private keys, scripts and multi-signature addresses.

II.2.3.1 Encrypted private keys

In managing private keys, we always seek to achieve two security objectives, which are confidentiality and availability. For this purpose, private keys must always remain secret and available for use. Currently, wallet applications use passphrase encryption to store the private keys in their database files. However, confidentiality can be missed when private keys are backed up in another media or when they imported or exported to another wallet. To overcome this security issue, a new

standard known as BIP-38 was devised to provide a common way for encrypting private keys with a passphrase and encoding them with Base58Check. It relies on the AES (Advanced Encryption Standard) algorithm, which was developed by NIST (US National Institute of Standards and Technology).

To produce a Base58check encrypted private key, BIP-38 scheme needs a WIF-encoded private key and a long passphrase. As long as the passphrase is strong, the encrypted private key will be strongly secure. The same passphrase is used to decrypt the encrypted private key when it is exported to another wallet. Note that a Base58check encrypted private keys always start with ‘6P’, while a WIF private key starts with ‘5’. Fig.14 illustrates the process of private key encryption as it is used by the BIP-38 Standard.

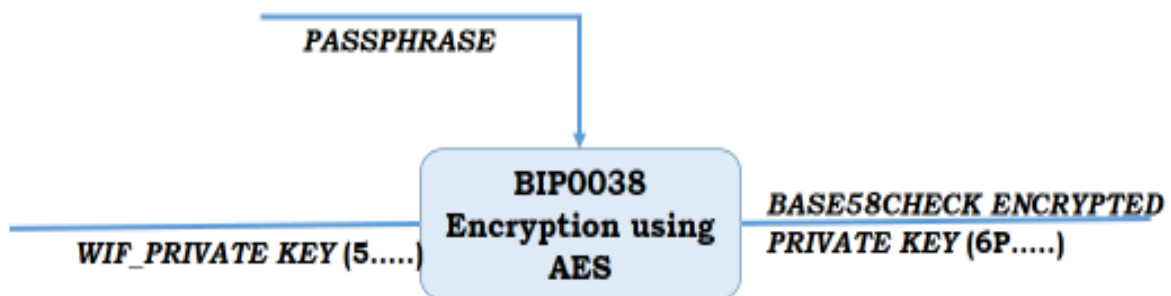


Fig.14 BIP-38-private key encryption

II.2.3.2 Pay-to-script addresses

Transactions made with traditional Bitcoin addresses, known also as pay-to-public-key-hash (P2PKH), can be spent by providing only the corresponding private key signature and the public key hash. However, sometimes a more than one signature is needed to spend funds. For that, BIP-16 introduced a new type of address, referred to as Pay-To-Script (P2SH) address, in which the beneficiary is a hash of a script instead of an owner of a public key. P2SH addresses are encoded with Base58check with a prefix version of ‘5’, which result in addresses that starts with ‘3’. They are created from a transaction script, which defines who can spend a transaction output [31].

P2SH addresses are not as the same as multi-signature addresses, but they often represent multi-signature scripts. They might also represent scripts encoding other types of transactions. More details on how the P2SH addresses work is provided later on in this paper (see transaction section). The following is an example of a P2SH address:

342ftSRCvFHfCeFFBuz4xwbeqnDw6BGUey [32]

II.2.3.3 Multi-signature addresses

A multi-signature address is made of a script that requires more than one signature to redeem the associated funds. It is considered as one of the common implementation of the P2SH. In an M-of-N multi-signature address, no funds can be spent at least M signatures were provided out of N signatures. More details is provided later on in this paper (see transaction section). In the next section we will examine different types of wallets and the way they implement and manage keys and addresses.

II.2.3 Bitcoin wallets

Unlike physical wallets that store coins and bills, digital wallets hold only keys and addresses to sign or unlock transactions. All the coins that a user holds in the system are stored on the Blockchain

in the form of transactions. Wallets are software applications that allow users to gain access to the Bitcoin peer-to-peer network so they can create, send or receive transactions to or from other users in the network. They generate and store private and public keys that users use to sign or unlock transactions. They also allow users to track their balances through transactions aggregation. Wallets exist in different forms, which can be categorized according their platform as follows (Fig.15):

- **Desktop wallet:** a desktop application that is used to send and receive bitcoins. It helps also the user to keep track of their balance. Transactions generated by this application are signed and checked using an ECDSA algorithm based on private and public keys.
- **Mobile wallet:** a mobile application that runs on smart-phone or tablet operating systems, for instance, Apple iOS and Android. They run a lightweight node and are simple and easy to use.
- **Web wallet:** they are web sites relying on a third party server that provides the same services as a stand-alone wallet. They allow the users to remotely manage their transactions.
- **Hardware wallet:** Bitcoin wallets that save the users' private keys in a hardware device. They are considered the best way to securely store large amounts of bitcoin.
- **Paper wallet:** they are Bitcoin keys printed in QR-code format on paper.

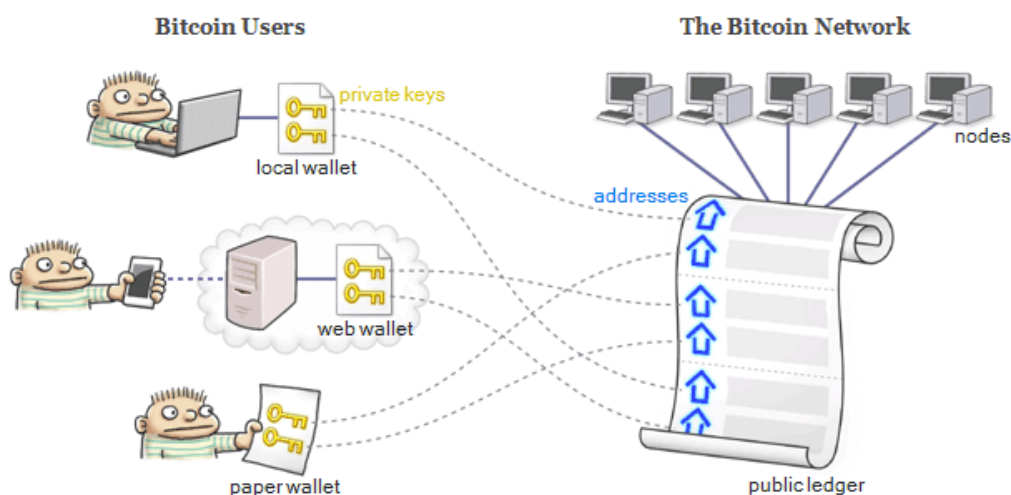


Fig.15 Some forms of Bitcoin wallets [33].

In Bitcoin, we distinguish two different families of wallets according to the way they generate keys and addresses. These are nondeterministic and deterministic wallets.

III.2.3.1 Nondeterministic wallets

Nondeterministic wallets, AKA type-0 wallets, were used in the first Bitcoin Core clients. They come with 100 pre-generated private keys and once all these keys are used, they start to generate randomly new keys. Since Bitcoin recommends using each key only once, many keys are created and backups are frequently needed. If backups are not made and the wallet crashes, the keys are lost and funds are lost too and there is no way to recover them. Nondeterministic wallets were very cumbersome to manage, hence they are being abandoned by Bitcoin community.

III.2.3.2 Deterministic wallets

Deterministic wallets, AKA seeded wallets, are an alternative way to manage keys that were introduced by the Bitcoin Improvement Protocol 32 (BIP32). Private keys originate all from a common seed. Seeds are presented in human-readable words such as a Mnemonic phrase. They are generated

randomly and are also combined with other data, known as the chain code, to derive the private keys. Knowing the seed is enough to recover all the derived private keys and therefore a single backup is sufficient. Table III suggests some advantages and disadvantages related to nondeterministic and deterministic wallets.

TABLE III PROS AND CONS OF NONDETERMINISTIC AND DETERMINISTIC WALLETS

TYPE OF WALLETS	ADVANTAGES	DISADVANTAGES
Non-deterministic wallet	generate randomly strong private keys	<ul style="list-style-type: none"> - cumbersome to manage, backup, and import - must keep copies of all the keys; - frequent backups are needed - funds may get lost if backups are not done and wallet crashes
Deterministic wallet	<ul style="list-style-type: none"> - Single backup is needed; - Its organizational structure help allocate a set of keys to each department. - Users can create a sequence of public keys without having access to the corresponding private keys 	Once the seed is discovered, keys can be determined and all the related transactions can be stolen.

The most commonly used form of deterministic wallets, which is described by BIP32, is the Hierarchical Deterministic Wallets, AKA type-2 HD wallets.

III.2.3.2.1 Hierarchical deterministic wallets.

These wallets are called hierarchical because they organize keys in a tree structure, parent keys that derive children keys and each children key derives many grandchildren keys, and so on, to an infinite limit (See Fig.16).

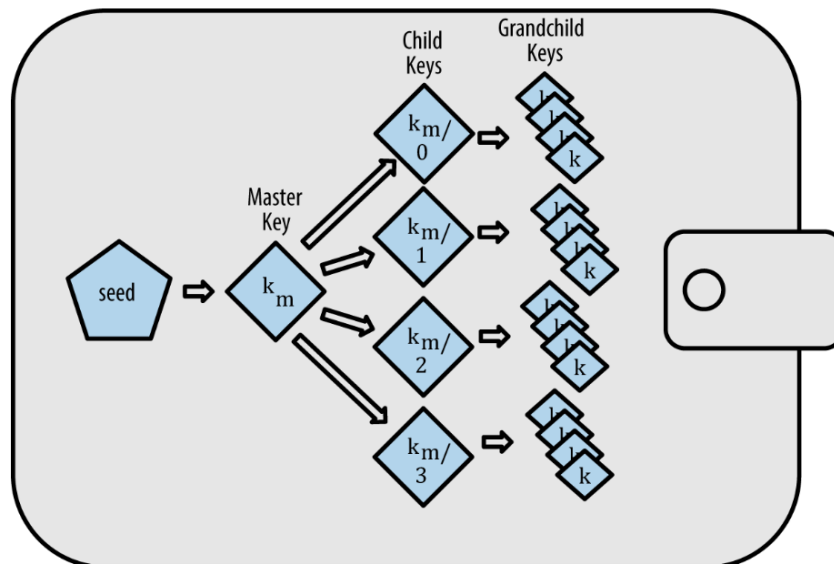


Fig.16 HD wallet- A tree of keys originating from one seed.

To make backups easier for average users, BIP-39 introduced a standardized way to create seeds from a sequence of English words. This mnemonic sentence is easy to remember, transcribe, and import or export across wallets. BIP-39 also implement the conversion of the mnemonic sentence into

a binary seed, which is used to generate the deterministic wallet. This standard is being adopted by many HD wallets.

Also and in order to enhance the wallets interoperability, security and flexibility, some standards of best practices have emerged in the crypto-currency realm. Some of these standards are as follows:

- BIP-39: Mnemonic code words
- BIP-32: HD wallets Implementation
- BIP-43: Multipurpose HD wallet structure
- BIP-44: Multicurrency and multi-account wallets

These standards are being implemented by many Bitcoin wallets such as Breadwallet, Copay, Multibit HD, Mycelium, Electrum, etc. In the following, we will examine some technical details related to these standards.

III.2.3.2.2 Bip-39- mnemonic code words

BIP-39 introduces the implementation of a mnemonic code words or mnemonic code sentence, which are a set of easy to remember English words forming a root seed. This seed is used for the generation of deterministic wallets as described in BIP-32 standards. BIP-39 consists of two parts: generating the mnemonic, and converting it into a binary seed. We will examine how mnemonics are created from an entropy and how seeds are generated from mnemonics.

III.2.3.2.2.1 Entropy to mnemonic sentence

BIP-39 uses a randomly picked number (also called entropy) between 128 and 256 bits that must be a multiple of 32 bits. Next, it adds a checksum to the picked number. The resulting number is divided into sections of 11 bits. The size of the checksum depends on the size of the picked number so that the resulting number is dividable by 11 (see Table IV). After that, each 11-bit value is linked to a word from the predefined list of 2048 words. Finally, the mnemonic sentence is made of these words. The whole process is depicted in Fig.17 with an example of an entropy of 128 bits.

TABLE IV RELATIONSHIP BETWEEN THE SIZE OF THE PICKED NUMBER, THE CHECKSUM, AND THE MNEMONIC SENTENCE.

Size of the random number (bits)	Checksum size (bits)	Resulting number size (bits)	Mnemonic length (words)
128	4	132	12
160	5	165	15
192	6	198	18
224	7	231	21
256	8	264	24

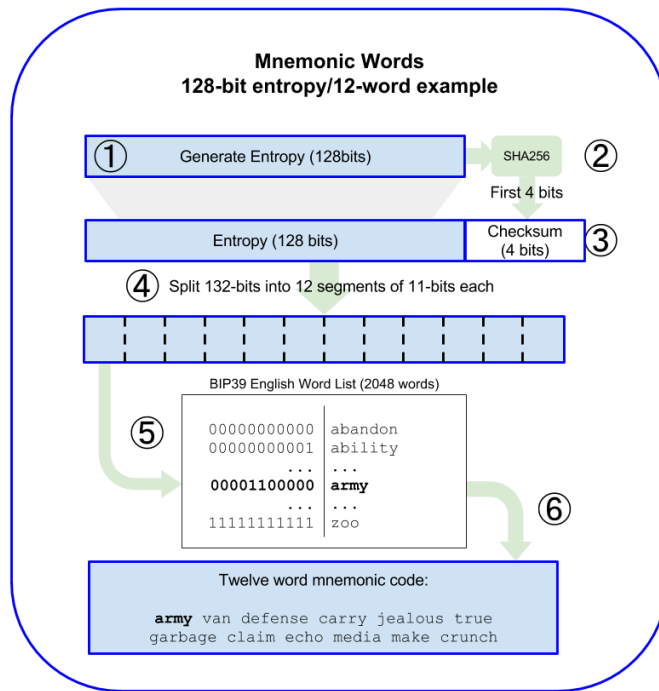


Fig.17 The process of generating a mnemonic sentence from a random number of 128 bits [34].

III.2.3.2.2 Mnemonic sentence to seed

To generate a strong and longer seed (512 bits), BIP-39 relies on a key-stretching function PBKDF2. This function was carefully designed to avoid brute force (too costly in computation) by using a salt and 2048 rounds of hashing. The salt consists of the mnemonic sentence concatenated with an optional passphrase provided by the user of the wallet to increase its security. The PBKDF2 relies on a HMAC-SHA-512 function that takes as entry the mnemonic and the salt. This function is run in 2048 rounds of hashing to produce a 512-bit seed. This process is well illustrated in Fig.18.

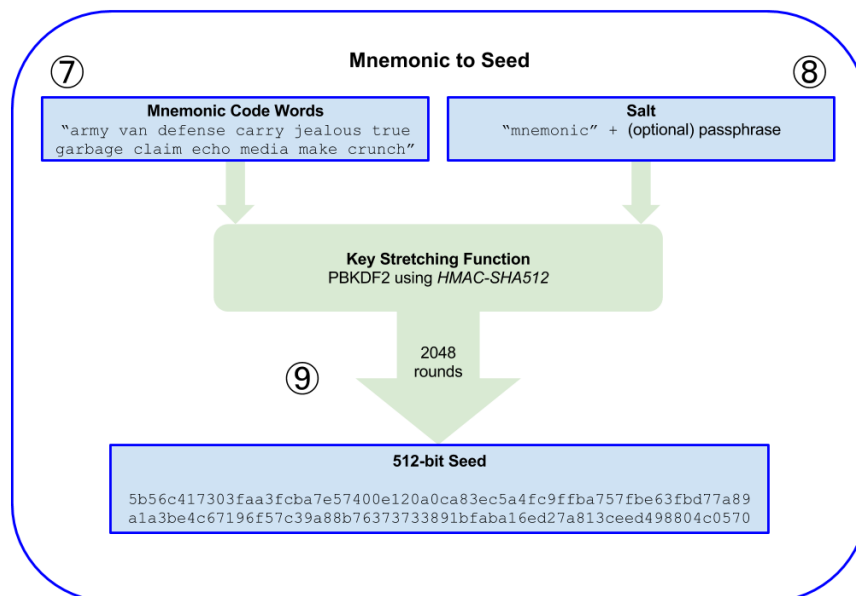


Fig.18 The process of generating a strong seed from a mnemonic sentence [35].

III.2.3.2.2 BIP-32- HD wallets implementation

BIP-32 describes the creating of an HD wallet from the seed. We will lay out some technical details about this process.

II.2.3.2.2.1 Creating an HD wallet from the seed

As previously mentioned, a HD wallet can be derived from a single root seed. This seed serves to generate the Master Private Key and the Master Chain code. The process starts with passing the root seed into hash function of 512 bits output, HMAC-SHA-512. The outcome is then divided into two parts: the 256-bit left part is taken as the Master Private Key (m), while the 256-bit right part is what forms the Master Chain code (c). The Master public key (M) is derived from the Master Private Key using the elliptic curve discrete logarithm $m \cdot G$. The chain code is used to provide some randomness in the creation of child keys from parent keys. Fig.19 illustrates this process.

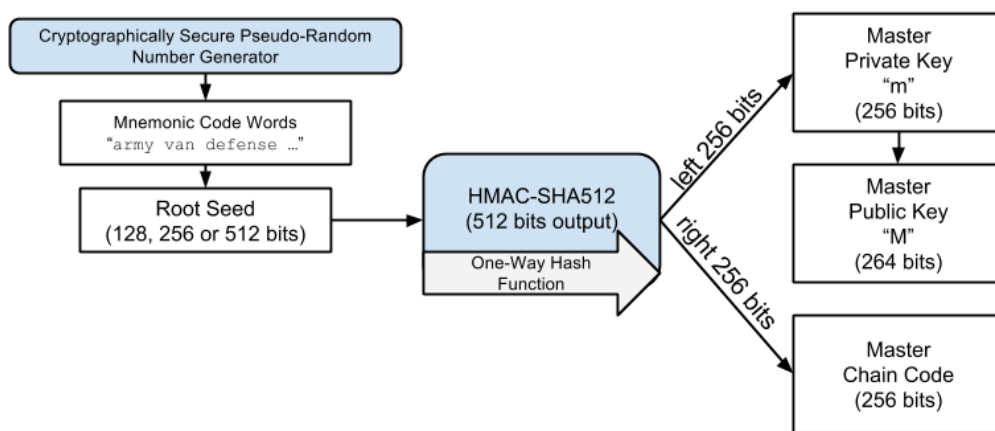


Fig.19 The process of creating the master keys and master chain code.

HD wallets use derivation functions to generate children keys from the master keys. We distinguish two type of functions: normal derivation functions and hardened derivation functions. To create new children, two elements are required: the parent key and the chain code. When these two elements are combined, they are called an extended key.

We distinguish private extended keys, which can generate private and public children keys, and public extended keys, which can only derive public children keys. Extended keys are 512 or 513 bits and are encoded using Base58Check, which make them start with “xprv” or “xpub”. We will examine in details these functions in the following sections.

III.2.3.2.2.2 Normal derivation functions

In the following sections we will examine two types of normal derivation functions, which are: private child key derivation and the public child key derivation.

III.2.3.2.2.2.1 Private child key derivation

Private Child keys are created from parent keys using a child key derivation (CKD) function. This function is based on a HMAC-SHA-512 Hash-function that takes as entry: a private key, a chain code, and an index of 32 bits (less than 2^{31}). It produces a 512-bit hash which is split into two 256-bit halves. The right-half of the produced hash is the child chain code. The left-half is concatenated with the parent private key to form the child private key. The index is a 32 bits number that start from 0. It allows each

parent key to create 2^{31} of normal children keys, and another 2^{31} of hardened children keys (see next section). The following flowchart (Fig.20) describes the steps of a CKD function.

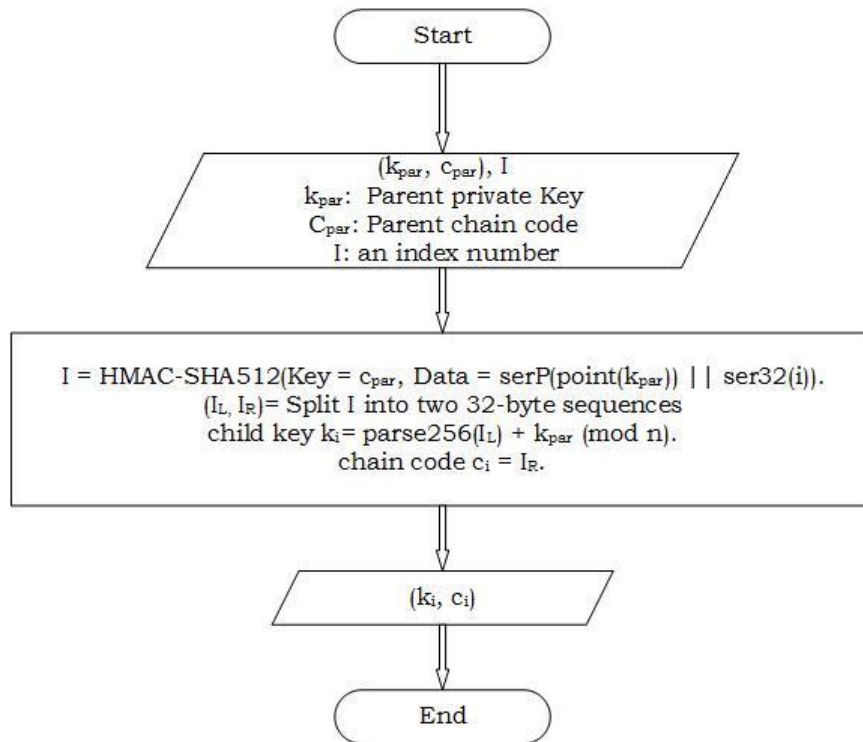


Fig.20 Child private key generation process.

The following functions are used in the CDK functions as described in the BIP-32 standard [36].

- $ser_{32}(i)$: serialize a 32-bit unsigned integer i as a 4-byte sequence, most significant byte first.
- $ser_P(P)$: serializes the coordinate pair $P = (x,y)$ as a byte sequence using SEC1's compressed form: $(0x02 \text{ or } 0x03) \parallel ser_{256}(x)$, where the header byte depends on the parity of the omitted y coordinate.
- $parse_{256}(p)$: interprets a 32-byte sequence as a 256-bit number, most significant byte first.

III.2.3.2.2.2 Public child key derivation

This function, called CKDpub, serves to create child public keys and child chain codes with using only an extended public key. It works only for non-hardened keys, which means keys with index numbers less than 2^{31} . This feature gives HD wallets the ability to derive public child keys from public parent keys, without any knowledge about the private keys. A common use case of the public extended key is to install it an e-commerce website where it is strongly needed to produce a new address for each online payment made.

Public child keys are derived from public parent keys, parent chain codes and an index numbers. All these elements are passed through a HMAC-SHA-512 which produce a public child key and a child chain code as shown in the following flowchart (Fig.21)

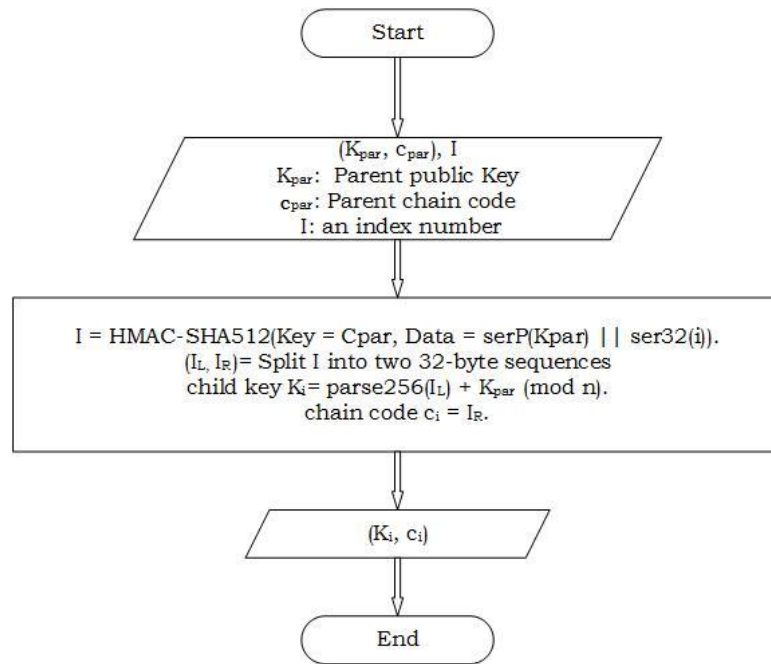


Fig.21. Derivation of non-hardened child public keys and child chain code from an extended public key.

Producing a chain code from an extended public key comes with a potential risk of compromising the HD wallet. If a child private key is disclosed or stolen, all the other child private keys can be revealed. Also, a child private key combined with a parent chain code can produce the parent private key, which may jeopardize the whole HD wallet and the related funds. This weakness was resolved through the use of hardened derivation functions.

III.2.3.2.2.3 Hardened derivation

Hardened derivation functions fixed a major weakness in HD wallets, where a chain code can be exploited with a private key to reveal other private keys. It suggests an alternative way to produce chain codes only from the parent private keys. This breaks any link between the parent public key and the child chain code. Fig.22. illustrates the hardened derivation, in which the public parent key is completely omitted and only the private key is used to derive the child chain code.

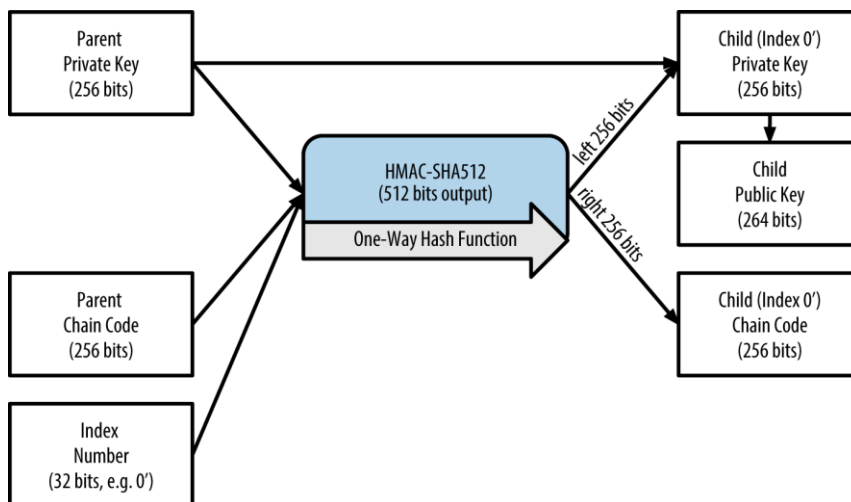


Fig.22 Hardened derivation of a child key [37].

Hardened derivation functions use index numbers between 2^{31} and $2^{32}-1$, while normal derivation functions use numbers between 0 and $2^{31}-1$. This makes it easy to distinguish between normal keys and hardened keys. Also, another way to display hardened indexes is to start from “0”, but using a prime symbol. Therefore, the hardened index “ 2^{31} ” would be “0’ ” and so on.

One of the best practices for HD wallets is the derivation of level-1 children of master keys using always a hardened derivation. This prevent from compromising the master keys.

III.2.3.3 Paper wallet

Paper wallets are Bitcoin private keys printed on paper. They may include the corresponding Bitcoin addresses too. Paper wallets are considered as an effective way to store backup offline, which is known also as cold storage. It is a secure way to avoid hackers, key-loggers, and other online threats. The major downside of paper wallets is that they are vulnerable to theft. Fig.23 illustrates a paper wallet that includes a Bitcoin address and an encrypted private key using BIP38 protocol.



Fig.23 Paper Wallet: a Bitcoin address and a BIP38-encrypted private key [38].

III.2.3.4 Brain wallets

Brain wallets are wallets that used a mnemonic phrase as a seed rather than a randomly chosen number. A hash-function (SHA-256) is used for this purpose with mnemonic passphrase as an input to generate the private key. The public key and the Bitcoin address are derived using the same techniques as for other wallets. Such seeds are generated by wallets like Electrum, Armory and Mycelium. Fig.24 shows a brain wallet generated using bitaddress.org with the passphrase: “Morocco is the Best Country in North Africa”

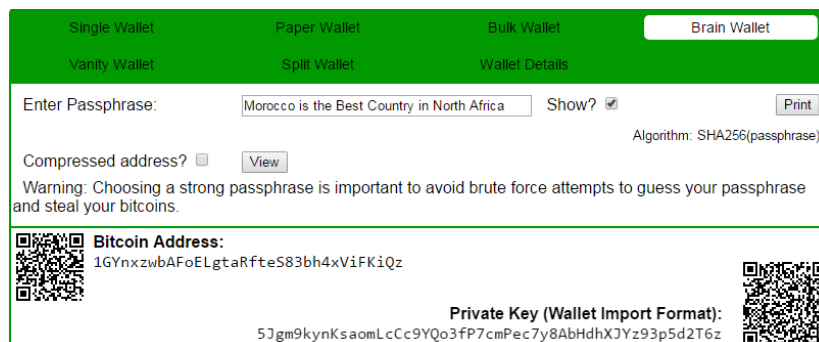


Fig.24 Brain wallet [39].

II.2.4 Bitcoin transaction

Bitcoin transaction is data structure that allows users to send and receive bitcoins. It consists of two main parts, which are the input and the output. Each transaction references another as its source of funds forming a chain through which a transaction spends the outputs of a previous transaction. The unspent transaction outputs are referred to as UTXOs and are tracked by the Bitcoin network. There were more than 54 million UTXOs in June 2017 [40]. UTXOs are locked for their owners using a locking script that relies on a specific scripting language. In order to redeem the value of bitcoins held by an UTXO, the participant should provide the corresponding unlocking script as a proof of ownership.

III.2.4.1 Transaction structure

A Bitcoin Transaction is built around two main parts, which are the input and the output. The output is the fundamental element since a transaction spends an output of another transaction. The following example (see Fig.25) illustrates a Bitcoin transaction structure with one input and two outputs.

```

{
  "version": 1,
  "locktime": 0,
  "vin": [ {
    "txid": "7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18",
    "vout": 0,
    "scriptSig"
    "3045022100884d142d86652a3f47ba4746ec719bbfd040a570b1deccbb6498c75c4ae24cb02204b9f039ff08df09cbe9f6addac960298cad530
a863ea8f53982c09db8f6e3813[ALL]
0484ecc0d46f1918b30928fa0e4ed99f16a0fb4fde0735e7ade8416ab9fe423cc5412336376789d172787ec3457eee41c04f4938de5cc17b4a10fa
336a8d752adf",
    "sequence": 4294967295
  } ],
  "vout": [ {
    "value": 0.0150
    "scriptPubKey": "OP_DUP OP_HASH160 ab68025513c3dbd2f7b92a94e0581f5d50f654e7 OP_EQUALVERIFY
OP_CHECKSIG"
  },
  {
    "value": 0.0845
    "scriptPubKey": "OP_DUP OP_HASH160 7f9b1a7fb68d60c536c2fd8aeaa53a8f3cc025a8 OP_EQUALVERIFY
OP_CHECKSIG",
  } ]
}

```

Fig.25 An example illustrating the structure of a real Bitcoin transaction [41].

The transaction outputs consist of two parts:

- Amounts of bitcoins:

* 0.0150 BTC (1,500,000 satoshis);

* 0.0845 BTC (8,450,000 satoshis).

- A cryptographic puzzle, encoded in a scripting language, as required conditions to spend the outputs. These are in the following example as:

```

* "scriptPubKey": "OP_DUP OP_HASH160 ab68025513c3dbd2f7b92a94e0581f5d50f654e7 OP_EQUALVERIFY OP_CHECKSIG"
* "scriptPubKey": "OP_DUP OP_HASH160 7f9b1a7fb68d60c536c2fd8aeaa53a8f3cc025a8 OP_EQUALVERIFY OP_CHECKSIG".

```

The transaction inputs refer to which UTXO to be spent (vout: 0 as the first UTXO of the transaction) and provide a proof-of-ownership in the form of an unlocking script, which is in the previous example as:

```

"scriptSig"
"3045022100884d142d86652a3f47ba4746ec719bbfd040a570b1deccbb6498c75c4ae24cb02204b9f039ff08df09cbe9f6addac960298cad530a863ea8f53
982c09db8f6e3813[ALL]

```

This unlocking script or *scriptSig* is most often a digital signature and a public key. The input indicates also the transaction ID, which reference the transaction that contains the UTXO being used. Note that Bitcoin transactions are sometimes formed with more than one UTXO. The *ScriptSig* is also known as a witness or witness data since it testifies to the true ownership of the funds being spent [42]. This witness data is being moved to another structure called segregated witness in order to prevent transaction malleability attacks.

Transactions are generally stored in an object-oriented structure, but transmitted over the internet in a serialized format, which allows transmission of one byte at a time. Most Bitcoin libraries have built-in functions to serialize and de-serialize transactions in order to convert transaction from to a byte-stream format and back to an object-oriented structure.

III.2.4.2 Transaction fees

Processing transactions implies fees that go the miner who found the proof-of-work of the new block. These fees, though very small, serve as incentive for the miners as a reward for their effort to secure the network. These fees are implied as the difference between the outputs and the inputs of a transaction. They give priority to the transaction to be processed ahead of those transactions that don't include fees. The cheapest recommended fee for fast confirmation is currently 330 satoshis/byte [43]. Since the average size of a transaction is around 226 bytes [44], this makes the average fees to be around 0.00075 bitcoins per transaction.

$$\text{AVERAGE FEES (in bitcoins)} = (226 * 330) / 10^8 \quad (4)$$

III.2.4.3 Transaction script language

This script language is used to write the unlocking and the locking script of the Bitcoin transactions. This language was intentionally designed to be limited in scope and execution for security reasons. It can be used to write locking scripts that express a wide range of complex conditions. The language does not include any loops, which prevents from creating infinite loops that could be injected in transactions and might cause denial of service attacks. Another interesting security feature is the language is stateless, which makes the execution of the scripts not requiring any prior state or saving a state after execution. This denies to hackers the use of the state property to influence the execution of the script.

III.2.4.3.1 Locking and unlocking scripts

In Bitcoin, locking scripts are part of every transaction output and most often appear as *scriptPubKey*, which represents conditions to be satisfied in order to redeem an output. In the other hand, an unlocking script is a script that fulfills some conditions placed on an output and allows the output to be spent. Unlocking scripts are found in every transaction input and are referred to as *scriptSig* or a *witness*.

Bitcoin nodes validate transactions by executing the locking and the unlocking scripts together. Since each transaction input contains an unlocking script and a reference to a previous UTXO, the node will retrieve the locking script from that UTXO and see if the unlocking script satisfies the locking script by executing them in sequence as shown in Fig.26. Once the conditions are met, the output in question is considered as spent and hence removed from the unspent transaction outputs (UTXO).

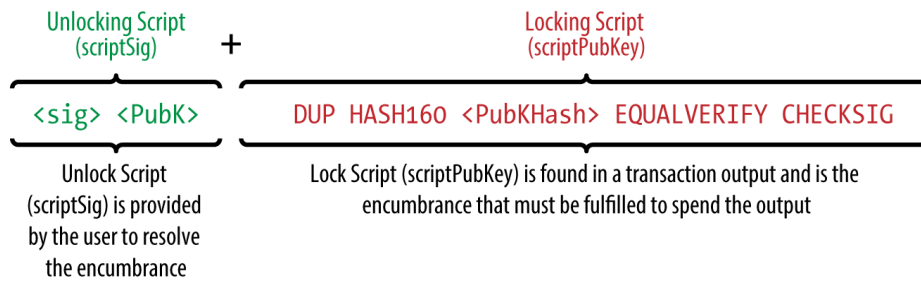


Fig.26 The concatenation of an unlocking script and a locking script.

For security reasons and since 2010, Bitcoin does not execute the unlocking and locking scripts in sequence, but they are instead executed separately. The unlocking script is executed first and then copied to the main stack and executed along with the locking script. If the result is TRUE then the transaction is valid.

III.2.4.3.2 The script execution stack

The scripting language that is being used in Bitcoin is known as a stack-based language since it relies on a data structure called a stack. A stack uses two main operations: push and pop. While push adds an element on top of the stack, Pop removes the top element from the stack. A stack proceeds as in LIFO (Last-In-First-Out) data structures.

Scripts are executed by processing items from left to right. Operators used to push or pop parameters to or from the stack, act on the parameters and in most cases push the result onto the top of the stack. For instance, OP_ADD will pop two items from the stack, add them, and push the resulting sum onto the stack [45]. Conditional operators evaluate a condition, producing a Boolean result of TRUE or FALSE. They are used in Bitcoin so they can produce the TRUE result, which validate the transaction. For example, OP_EQUAL pops two items from the stack and pushes TRUE (represented by 1) if they are equal or FALSE (represented by 0) if they are not equal.

III.2.4.4 Pay-to-public-key-hash (P2PKH)

Most often, transactions are sent to Bitcoin addresses in the form of a Pay-to-Public-Key-Hash (P2PKH), in which the locking script locks the output to a public key hash. This output can be unlocked by a P2PKH script in the form of a public key and a digital signature generated using the corresponding private key. The Locking script of a P2PKH transaction is in the form of:

OP_DUP OP_HASH160 <Public Key Hash> OP_EQUALVERIFY OP_CHECKSIG

The Public Key Hash is equivalent to the Bitcoin address, without the Base58Check encoding. The unlocking script is in the form of:

<Signature> <Public Key>

When combined, the two scripts would form the following validation script:

<Signature> <Public Key> OP_DUP OP_HASH160 <Public Key Hash> OP_EQUALVERIFY OP_CHECKSIG

The step-by-step execution of the validation script is illustrated in Fig27 (See Appendix II).

III.2.4.5 Digital signatures (ECDSA)

Bitcoin relies on digital signatures to ensure the proof of ownership of private keys. It uses the Elliptic Curve Digital Signature Algorithm (ECDSA), which is based on public/private key pairs. This algorithm is implemented in the script functions such as OP_CHECKSIG, OP_CHECKSIGVERIFY, etc. The ECDSA consists of an algorithm for making a signature using a private key and a transaction

or parts of it, and another algorithm to verify the signature using the public key and the transaction or parts of it.

III.2.4.5.1 The signing algorithm

The signing algorithm used in Bitcoin is as follows:

$$\text{Signature} = \text{Signing_Function}(\text{Hash_Function}(\text{Message}), \text{prvKey})$$

Where:

- Message is the transaction or parts of it (precisely a Hash of its subset)
- prvKey is the private key
- Hash_function is the hashing function
- Signing_Function is the signing algorithm
- Signature is the produced signature

The signing function produces a signature composed of two values: R and S.

$$\text{Signature} = (R, S)$$

Once calculated, these two values R and S are serialized into a byte-stream using a standardized encoding scheme, known as the Distinguished Encoding Rules (DER).

The signing function uses a temporary and a random private k to produce a temporary public key, Pk ($Pk=k*G$). This random pair of keys k/Pk is used in the calculation of the R and S values (see Fig.28 in Appendix III). R is x coordinate of the temporary public key.

$$R = X_{Pk}$$

The other value, S, is calculated using the following formula:

$$S = k^{-1} (\text{Hash}(m) + \text{prvKey} * R) \text{ mod } p \quad [46] \quad (5)$$

where:

- k is the temporary private key and k^{-1} is its inverse
- R is the x coordinate of the temporary public key (Pk)
- prvKey is the signing private key
- m is the transaction data
- p is the prime number used in the elliptic curve, defined in SECP256K1.

III.2.4.5.2 Verifying algorithm

The verifying algorithm is practically the inverse of the signature generation function. It uses the R, S values and the public key to calculate a point on the elliptic curve P, which is the temporary public key Pk used in the signing function (see Fig.29 in Appendix IV).. It is based on the following formula:

$$P = S^{-1} * \text{Hash}(m) * G + S^{-1} * R * \text{PubKey}. \quad (6)$$

where:

- R and S are the values produces by the signature
- PubKey is public key
- m is the transaction data, used in the signature
- G is the elliptic curve generator point, defined in SECP256K1.

If $X_P=R$ then the signature is valid. If otherwise, the signature is invalid.

III.2.4.5.3 Multi-Signature

Multi-signature transactions, known also as M-of-N transactions, include N public keys in their script and M of those must sign to unlock the funds. The maximum number of signatures is set to 15 in order to avoid producing transactions of big size. These type of transactions have a locking script in the form of:

$M <pubKey 1><pubKey 2> \dots <pubKey N> N CHECKMULTISIG$

Where:

- CHECKMULTISIG is a function that verify multi-signatures;
- N is the total of public keys used in the transaction
- M is the required signatures to unlock the funds.

The unlocking script for these kind of transactions look like:

$<Signature 1><Signature 2> \dots <Signature M>$

Combing the unlocking and the locking script would produce the following validation script:

$<Signature 1><Signature 2> \dots <Signature M> M <pubKey 1><pubKey 2> \dots <pubKey N> N CHECKMULTISIG$

III.2.4.5.4 Pay-to-script-hash (P2SH)

These kind of transactions were introduced in order to lessen the complexity of some transactions, especially those with many required signatures. It has the advantage of making complex scripts look as easy as a payment to a Bitcoin address. Complex scripts, known as redeem script, which hold the required conditions for spending the funds are replaced with a simple hash in the locking script. The redeem script is presented as part of the unlocking script when the UTXO is spent. Table V illustrates an example of a P2SH.

TABLE V LOCKING, UNLOCKING, AND REDEEM SCRIPTS OF P2SH

TX CONTENT	EXAMPLE
Redeem Script	$M <pubKey 1><pubKey 2> \dots <pubKey N> N CHECKMULTISIG$
Locking script	HASH160<hash of the redeem script (20 bytes)> EQUAL
Unlocking script	$<Signature 1><Signature 2> \dots <Signature M><redeem script>$

Another important feature of P2SH is to encode a script hash as a Bitcoin address, making these kind of payment easier for average users. P2SH addresses use the version prefix “5” and are encoded with Base58Check encodings, which produces addresses that start with “3”.

III.2.4.6 Transaction time-lock

In Bitcoin, transactions have a sort of postdating feature called the time-lock or nlocktime. Most of transactions have this feature set to zero, which means immediate dissemination and execution. When this attribute is set to a value less than 500 million, it is understood as a block height, which means it will be included in the block that has this specified height. If it is bigger than 500 million, it is interpreted as a timestamp, a Unix Epoch timestamp (starting date is Jan-1-1970).

III.2.4.7 Transaction security issues

Chained transactions that are created and broadcasted at the same time may not arrive in the correct order. To handle this issue, Bitcoin system stores temporary the child transactions in the orphan

transaction pool until the arrival of their corresponding parent transactions. To prevent any exploit of this pool by DDOS attacks, Bitcoin has defined a limit as MAX_ORPHAN_TRANSACTION to not exceed by any node. This solution solved the vulnerability that affected Satoshi Bitcoin Clients, referred to as CVE-2012-3789 [47].

Other interesting security features, embedded at the transaction level, are represented by the stateless and limited script language that is used in the locking and the unlocking script. The stateless property makes the execution of the scripts not requiring any prior state or saving a state after execution. This deny to hackers the use of the state property to influence the execution of the script. Also, the language does not include any loops, which prevents from creating infinite loops that could be injected in transactions and might cause denial of service attacks.

Bitcoin system involves a specific type of transaction, known as the Data output or OP_RETURN. This type of transaction allow users to store 40 bytes of hashed data in the Blockchain. OP_RETURN empowers the Blockchain to be used beyond the payment system to include more other applications, such as notarization, proof of existence, smart contracts, and so on. However, malevolent users could use this feature as a point of attack to flood the Blockchain with random data and increase the size of the ledger as a consequence. If this happens, it will be a burden for the users who are running full nodes. This risk and others are examined in details in the risk assessment section.

II.2.5 Bitcoin network

Bitcoin works over the internet using a peer-to-peer (P2P) network of equal nodes, which are connected to each other in the form of a mesh network. This network is completely decentralized, which means that there is no server that controls the system. Bitcoin adopted this kind of network because its resiliency and decentralization, and also its success in file sharing as used in BitTorrent. Bitcoin network refers to the set of nodes running the Bitcoin peer-to-peer network, while the extended Bitcoin network refer to the whole network which includes the Bitcoin peer-to-peer network, the pool-mining protocols, and other protocols.

Despite that peer-to-peer network nodes are equal, they take different roles related to the functionality they are playing. We distinguish four types of functions: network routing, Blockchain database, wallet services, and the mining (see Fig.30).

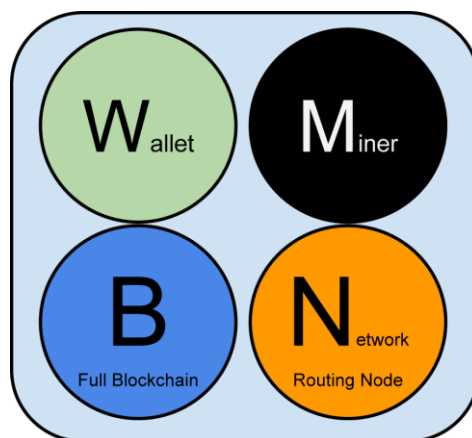


Fig.30. The four types of functionalities played by Bitcoin nodes [48].

In Bitcoin, all the nodes play the network routing function, which includes validating, disseminating transactions and blocks, discover and maintain connections to other nodes. Blockchain nodes are nodes that contain a copy of the distributed ledger. We distinguish two categories of Blockchain database nodes: one is called full node since it contains a full copy of the ledger and can serve to independently verify transactions, the other is called lightweight node and can verify transactions using a method known as simplified payment verification (SVP). A light-weight node, AKA SVP node, contains a subset of the Blockchain, mainly the headers of all the blocks that form the Blockchain. Wallet services are nodes that serve to send and receive transactions. They run either full nodes or lightweight nodes. Mining services are nodes that compete against each other or work with each other in order to solve the cryptographic puzzle as a requirement to add a new block to the ledger. The solution to the cryptographic puzzle is known as the proof-of-work. Solo miners use full nodes, while the mining pool nodes use only a light-weight copy of the ledger and rely on pool servers to verify the transactions. Fig.31 illustrates various types of nodes in an extended Bitcoin network (See Appendix V).

III.2.5.1 Network discovery

A new node must discover and connect to a minimum one node on the P2P network. For this purpose, it uses a TCP connection to port 8333 and start a handshake in which it propagates a version message. This version message consists of the version of the client software being used for the connection (nVersion), the current time (nTime), its Ip Address (addrMe), the remote node IP address (addrYou), its Block height (BestHeight), etc. the connection is established only if the peers versions are compatible.

To find peers of the P2P network, a node relies on DNS seeds. These DNS seeds are servers that provide a list of addresses of Bitcoin nodes. Once a node is connected, its neighbors will send its IP address to their neighbors helping this newly connected node to be well connected.

III.2.5.2 Full nodes

Full nodes hold a full and up-to-date copy of the Bitcoin ledger, which contains all the validated transactions. They can independently verify transactions without relying on any other node. They receive newly mined blocks through the network, which they add to their local copy of the ledger. Full nodes requires storage space of disk. The current size of the Blockchain is more than 141,331 MB [49].

III.2.5.3 Simplified payment verification (SVP) nodes

Most of the devices used in Bitcoin such as tablets and smartphones are resource-constrained, which makes them unable to run a full Blockchain node. For these devices, a method called simplified payment verification is used allowing them to operate without storing a full copy of the ledger. SVP nodes store only the block headers and any of the transactions. This cut down the storage size by 1000 times but deny the SVP nodes the ability to construct a full picture of the available UTXOs in the network.

II.2.6 Bitcoin's Blockchain

The Blockchain is a distributed database that stores the transactions that ever happened in the Bitcoin system. These transactions are stored in different blocks that are linked to each other. Each block references to its previous one forming a chain of blocks that goes all the way back to the first block ever created, AKA the genesis block (see Fig.32).

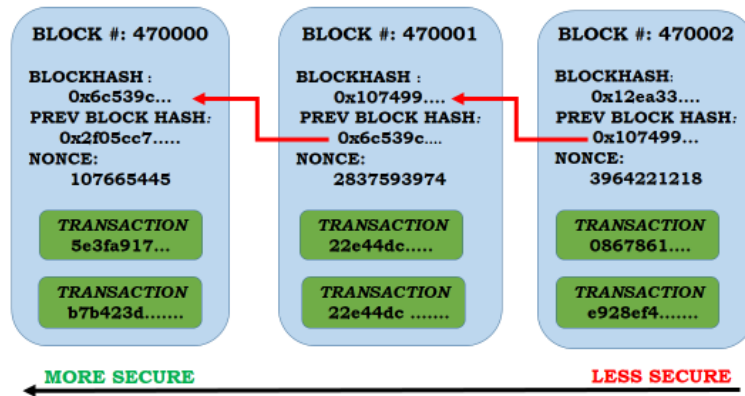


Fig.32 Chain of blocks forming the Blockchain.

This back-linkage of blocks is what makes the Blockchain secure and hard to corrupt or alter. To better illustrate this, let's suppose that a hacker with a big computational power wanted to change the transactions that are in block 470000. If he wants to be successful he has to calculate a new proof-of-work of that block, which takes around 10 minutes and has to catch up with the network since another block will be mined in the same period of time. Also, because of the back-linkage, the attacker must recalculate the new proof-of-work of block 470001, and the blocks that come after it. For this reason, it is considered that after 6 blocks mined, the transactions would never change even with more than 51% of the current computational power.

Orphaned blocks are valid blocks which are not part of the Blockchain. They occur when two or more miners produce blocks at the same time. This is also known as the fork. It can also be caused by an attacker with enough computational power as an attempt to reverse transactions. Bitcoin has reached a record number of orphaned blocks in June 2016, which was estimated around 40,000 blocks [50]. To solve the orphaned blocks or the fork issue, Bitcoin system considers the longest chain of blocks as the valid Blockchain (see Fig.33).

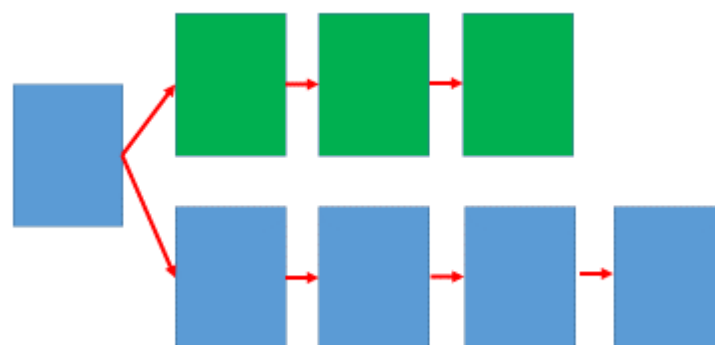


Fig.33 Bitcoin fork-the longest chain is the valid Blockchain.

The Blockchain size is soaring at a linear rate. It has doubled since the last year. In June 2016, it was around 70 GB and now it is around 120 GB on June 18th, 2017 [51]. Fig.34 illustrates this rapid increase in the ledger's size. We can infer that the size of the Blockchain will continue to double every

year and it might reach a size of 1TB by June 2020. This increase will be a real burden for users running full nodes and solutions should be devised to overcome this encumbrance.

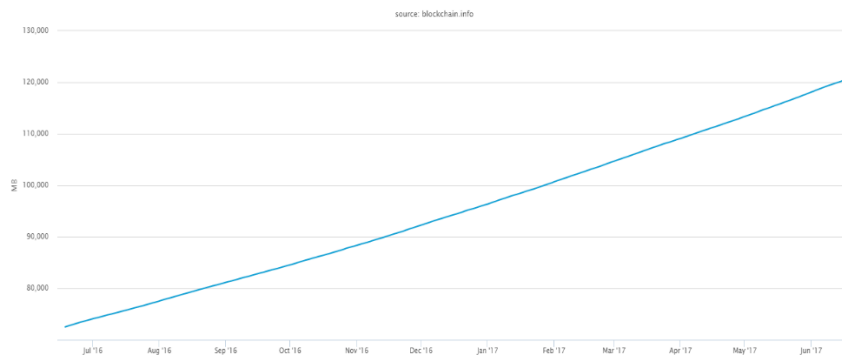


Fig.34 the Blockchain increasing size [52].

II.2.7 Bitcoin’s block

II.2.7.1 Block structure

The block is a data structure that holds a set of transactions that are included in the Blockchain. Its structure is made of a header and a collection of transactions (Fig.35). The block is identified by its position in the Blockchain, which is called the height. It is also identified by its cryptographic hash, which is double hash of its header using a SHA-256 hash function. While the header’s hash suffices as a unique identifier for a block, collision could happen and we might end-up with the same hash for two different blocks and this could create an integrity issue for the Blockchain database. A better way to prevent this issue from happening was to identify any block by its height and its header’s hash.

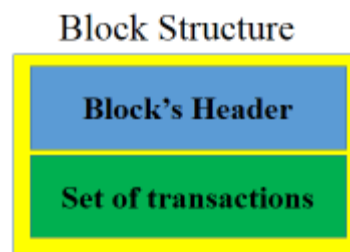


Fig.35 Block structure

II.2.7.2 Block header’s structure

The block header is what miners use for the proof-of-work finding. It contains the previous block hash, which makes blocks chained to each other (see Table VI). This force anyone who want to alter the previous block to redo the proof-of-work of this block and the other blocks that come after it. This security measure helps preserving the block’s integrity. In addition, the use of the Merkle tree, which is digital footprint of all the transactions included in the block, serves a way to ensure the block transactions’ integrity. This means that anyone who want to change, add or delete a transaction, have to recalculate the Merkle root and redo the proof-of-work of that block and the proof-of work of all the blocks that come after it.

TABLE VI: BLOCK HEADER STRUCTURE.

BLOCK HEADER PROPERTY	SIZE
Version (of software/protocol)	4 Bytes
Previous Block Hash	32 Bytes
Merkle Tree	32 Bytes
Timestamp	4 Bytes
Difficulty	4 Bytes
Nonce	4 Bytes

The version attribute indicates a version number to help keep track of the software/protocol upgrades, while the Previous Block Hash is a reference to a previous block hash, which connects this block to the previous block in the Blockchain.

The Merkle tree is the footprint of all the transactions included in the block. It is used to check the integrity of the transactions in order to avoid any alteration in the block transactions. It double-hashes pairs of transactions in different levels by combining the result in pairs and hashing them again. This process is continued until reaching the double-hash of the last pair, which is called the “Root Hash”. If the number of the block’s transactions is odd, the last transaction is duplicated. Fig.36 depicts the process of creating the Merkle tree of a block’s transactions

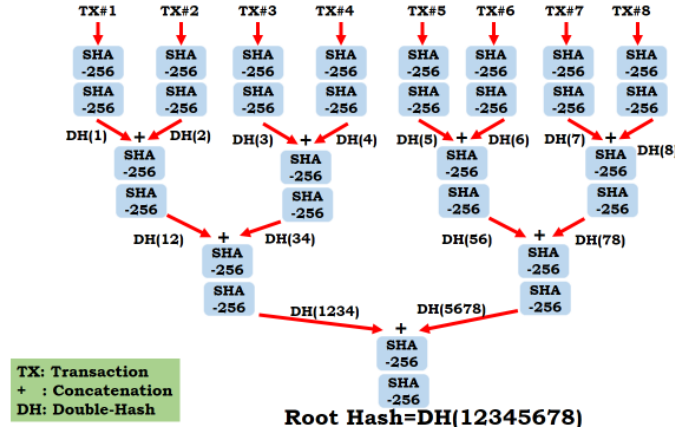


Fig.36 The process of creating a Merkle tree of a block of transactions.

II.2.8 Proof of work and mining process

Miners validate new transactions, add them into a new block, and race using their computational power to find the proof-of-work of this new block in order to store it in the Blockchain. The miner who wins the race is rewarded with brand-new bitcoins and transactions fees.

The proof-of-work serves as proof that the miner has committed a great amount of hashing power to find the block header’s hash that satisfies the required condition. The proof-of-work is hard to find but easy to verify. It involves finding a value for the nonce that results in a block’s header hash, using SHA-256 algorithm, that is less or equal to the difficulty target (target). So how this target is calculated?

Every block contains a field called “Bits”, known also as target Bits, is a four-byte number represented in a hexadecimal floating point format. Bits value serves to calculate the difficulty target, which is used as a condition in the mining algorithm. As shown in Fig.37, the Bits field value of the first block in the Blockchain is 1d00ffff. By convention, the first two digits (1d) represent the total number of digits a target is made of. It is used in the exponent of the floating point notation while the remaining digits (00ffff) represents the coefficient.

To calculate the target from the Bits value, we rely on the following formula:

$$\text{TARGET} = \text{COEFFICIENT} * 2^{(8 * (\text{EXPONENT} - 3))} \quad (7)$$

Using the hexadecimal representation and applying this formula to the block #0 with Bits value of (0x1d00ffff), the target would be:

$$\text{TARGET} = 0X00FFFF * 2^{(0X8 * (0X1D - 0X3))}$$

Therefore, the result in hexadecimal format is:

TARGET (in HEX) = 0xffff00

Block #0

BlockHash 00000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

Summary	
Number Of Transactions	1
Height	0 (Mainchain)
Block Reward	50 BTC
Timestamp	Jan 3, 2009 6:15:05 PM
Mined by	
Merkle Root	4a5e1e4baab89f3a32518a88c31bc8...
Difficulty	1
Bits	1d00ffff
Size (bytes)	285
Version	1
Nonce	2083236893
Next Block	1

Fig.37. Bits value field of the block #0 [53].

Now let’s compare the header’s hash of the Block #0 with the calculated target, using python 3. Script 1 shows that the Block Header’s Hash is less or equal the calculated target.

```
>>> #calculating the target of Block #0 using the Bits Value
>>> Target = 0x00ffff*2**(0x8*(0x1d-0x3))
>>> # the decimal number is:
>>> print (Target)
26959535291011309493156476344723991336010898738574164086137773096960
>>> # the Target in hexadecimal representation
>>> hex(Target)
'0xffff000000000000000000000000000000000000000000000000000000000000'
>>> # Now let's compare this target to the block #0 header's hash
>>> BlockHash=0x00000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
>>> BlockHash <= Target
True
```

Script 1. Calculating difficulty target from Bits value.

The target condition sets the frequency at which a new proof-of-work is found. It determines also the difficulty for a collection of blocks. Since the computational power is increasing at a rapid speed and the Bitcoin network must keep the block generation time at 10 minutes in average, the target should adjust accordingly. The retargeting is happening dynamically on every full node independently for every 2016 blocks, which occurs every 2 weeks. The retargeting formula used by Bitcoin full nodes is:

$$\text{New Target} = \text{current Target} * (\text{time on minutes of the last 2016 blocks}) / 20160 \text{ minutes.} \quad (8)$$

rate. It shows also that difficulty and the hash rate have tripled between June 2016 and the same month of 2017. This is due mainly to competition between miners.

TABLE VII. DIFFICULTY AND HASH RATE CHANGE BETWEEN 2016 AND 2017 [55].

Date	Difficulty	Hash Rate (GH/s)
June 17 th , 2017	711,697,198,174	5,094,526,985
June 21 st , 2016	209,453,158,595	1,499,324,110
Ratio of change	3.397882385	3.397882386

Without the difficulty, any miner possessing big hashing power would take over the Blockchain and could change it at will, therefore the difficulty participates strongly to the security of the Bitcoin.

With high difficulty and competition, miners went from using CPU/GPU mining to ASIC mining. ASIC, Application-Specific Integrated Circuits, are machines in which the mining algorithm is hard-coded directly in the chips. They hold the advantage of processing speed (Fig.40).



Fig.40 ASIC machine [56].

To increase their computational power, miners use several ASIC machines in big warehouses. This process consumes a lot of electricity and produce a huge amount of heat, which makes cooling systems run continuously adding more consumption of electricity. Fig.41 shows a mining warehouse made of ASIC machines.



Fig.41 Mining warehouse made of ASIC machines [57].

These machines run a proof-of-work algorithm that is hard coded in their chips. After validating a set of transactions, calculating the new target Bits, and forming a header of the candidate block, the mining node runs the proof-of-work algorithm. This algorithm searches for hash of the candidate block concatenated with a nonce that is less or equal the target. The following script (Script 3) illustrates an example of the proof-of-work algorithm.

```

- proof-of-work(header, bits)
- starting_time= getTime()
- target=calculate_target(bits)
- max_nonce= 232 # 4 billion of possible iterations
- for nonce in range (max_nonce):
    Hash_result= SHA256(header+nonce)
    If Hash_result<=target:
        Ending_time=getTime()
        timeOfMining= Ending_time- starting_time
        Return (Hash_result, nonce, timeOfMining)
    Print("Unsuccessful mining")
Return(nonce)

```

Script 3: Proof-of-work algorithm

Mining is a costly activity and it is similar to gambling. Despite the ASIC mining machines, miners are not sure to find a proof-of-work and may lose electric power without a compensating reward. For this reason, Miners now collaborate to form mining pools, pooling their hashing power and sharing the reward among thousands of participants. This could rise a new threat to the Bitcoin security. We are calling this threat, mining pools hijacking.

Fig.42 illustrates that only five pools hold more than 57% of the hashing power. These largest mining groups are: AntPool, BTC.TOP, BTCC, Bixin, and BTC.com. To ensure the Bitcoin security, these mining pools have to stay honest since they detain a huge computational power.

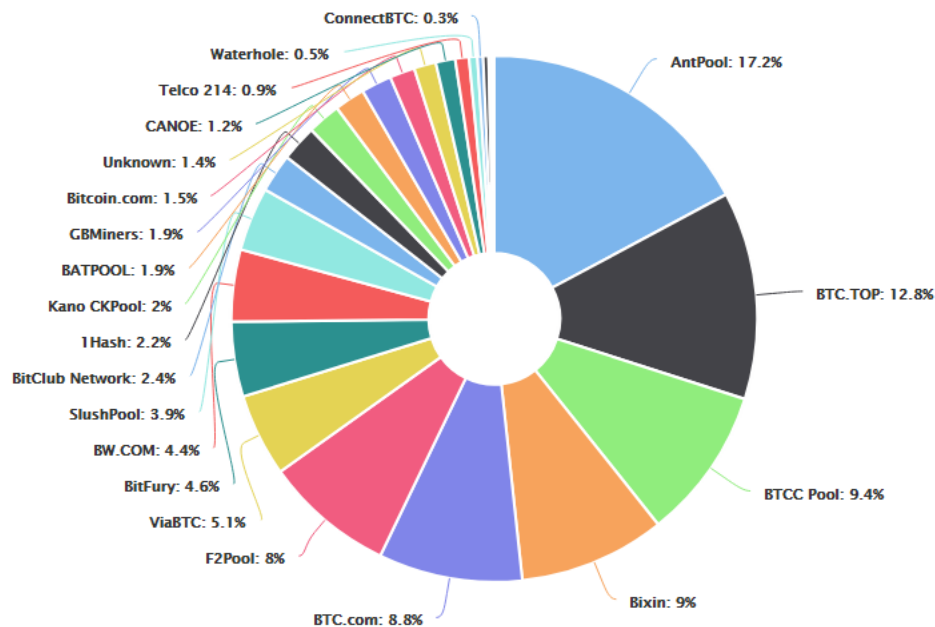


Fig.42 Hashrate distribution amongst the largest mining pools [58].

All the aforementioned concepts suggest that Bitcoin is a secure by design crypto-currency. Its security relies on the cutting-edge cryptographic technologies such as the digital signature and the hash

functions. However its security is tightly linked to an underlying assumption, which claims that the miners will always stay honest and work for the security and never against it since there is a financial incentive. In the following sections we will go over some risk assessment methods that are currently used to evaluate the security of information systems. Later on in this paper, we will assess the Bitcoin security using one of these methods.

For further information about how Bitcoin works, you can refer to the amazing Book of Andreas Antonopoulos entitled “Mastering Bitcoin, Second edition of 2017” [59].

II.3 Risk analysis

This section provides an overview of the risk analysis methods used to assess security for information systems. In addition; it outlines steps of the EBIOS method.

II.3.1 Risk assessment methods for information security

Information security is defined as the preservation of confidentiality, integrity and availability of information; in addition, other properties such as authenticity, accountability, non-repudiation and reliability can be involved [60]. This information could be represented in different forms such as paper or electronic devices. Managing information security is a big challenge with the ever-growing threats in the cyber-security realm.

Sustainable organizations assess continuously the risks related to their businesses and the information they rely on to keep their competitive advantage through the use of a risk management process. This process involves four activities, such as risk assessment, risk acceptance, risk treatment, and risk communication.

Risk assessment is the cornerstone step in the risk management process. It involves two techniques, which are: risk analysis and a risk evaluation. While risk analysis aims to identify possible sources of risk, threats or events with harmful impact along with their probability of occurrence, risk evaluation helps determine the significance of the identified risks by comparing them with a set of risk criteria.

There are several information security risk assessment methods available for use. Although they come with different cost and complexity, they tend to achieve the same purpose, which is the analysis and the evaluation of risks pertaining to information security. These methods are but not limited to: MEHARI (CLUSIF, 1997), OCTAVE (CERT, 1999), CRAMM, IRAM, EBIOS (ANSSI, 1995), MAGERIT, etc. There are also some guidelines that address the same issue, such as ISO 27005, NIST SP800-30, security risk management guide, Australian IT security handbook, etc.

II.3.2 Some risk management methods

There are many Risk management methods that help organizations and security professional study the risks related to information technology. Some of these methods are:

- **EBIOS** is a French method, invented in 1995 by ANSSI (French National Agency for Information Systems Security) and it is being used by many private and public institutions around the world. Straight and rigorous, this method is a reference in the private and public sector in France and abroad. In addition, many organizations uses it to conduct their own ISS risk analysis.

- **MEHARI** (CLUSIF, 1996), is an integrated and a comprehensive ISS risk assessment and management method, first developed by the French Information Security Club, known as CLUSIF in 1996. It is based on a quantitative approach and is now being diffused and developed by Quebec Information Security Club, known as CLUSIQ [61].

- **OCTAVE** (CERT, 1999), created mainly for the U.S. Department of Defense (DoD) to help address their information security challenges, it is mostly used to assess an organization's information security needs and can be tailored to fit the organization's unique risk environment, security and resilience objectives, and skill level [62];

- **CRAMM** (CCTA, 1985), a risk analysis method developed by the British government organization CCTA (Central Communication and Telecommunication Agency) which is now renamed

the Office of Government Commerce (OGC), it is rather difficult to use without the CRAMM tool [63].

All these methods help achieve the same purpose, however we are using a very straightforward method developed by ANSSI. We describe this method in the following section of this paper.

With all this diversity of methods and guidelines, choosing a method to conduct information security risk assessment may seem challenging. Risk assessment is a resource-consuming task in terms of time, expertise and people involved. A pertinent choice of the appropriate method would save time and frustration. A comparative study done by Filipe Macedo and Miguel Mira Da Silva ranked methods such as OCTAVE, EBIOS, MAGERIT, IRAM, IT-Grundschutz, and MEHARI as the most relevant methods in terms of moderate complexity, structured approach, and available tools [64].

All the previously mentioned methods, though differ in complexity and scope, could be used to carry out a comprehensive risk assessment of Bitcoin security and could lead almost to the same conclusions. The main reason we are using EBIOS is because of its flexibility and adaptability. The following section describes EBIOS risk assessment method.

II.3.3 EBIOS risk assessment method

EBIOS stands for “Expression des Besoins et Identification des Objectifs de Sécurité”, which is translated in English as “Expression of Needs and Identification of Security Objectives”. It is a free and comprehensive risk assessment method, invented in 1995 by the French National Agency for Information Systems Security (ANSSI). It is currently supported by a non-profit association of risk management experts, known as EBIOS Club. EBIOS is used by many private and public organizations, in France and abroad, to conduct information systems security (ISS) risk analysis. It helps also produce different security documents such as the security master plan, security policy, protection profile, risk mapping, etc. It is adaptable to different security contexts and can be applied to either basic or complex systems. Furthermore, EBIOS is compliant with major IT security standards [65], such as:

- ISO/IEC 27001: A standard that provides requirements for an information security management system (ISMS).
- ISO/IEC 15408/15443: Evaluation criteria for IT security, known also as common criteria.
- ISO/IEC 17799: Code of best practices for information security management
- ISO/IEC 13335: Management of information and communications technology security. It has currently a status of withdrawn in ISO website [66].
- ISO/IEC 21827: Systems Security Engineering Capability Maturity Model.

EBIOS defines risk as a scenario in which risk sources exploit vulnerabilities on the supporting assets which cause incidents for the primary assets. The level of risk is estimated in terms of severity (gravity) and likelihood. Severity is defined as the magnitude of a risk, depends on the level of identification of personal data and the level of consequences or the potential impacts. The likelihood is the feasibility of a risk to occur and it depends on the level of vulnerabilities in the supporting assets.

EBIOS involves a five-stage of an iterative approach as illustrated in Fig.43. The first phase, known as the context study, aims to identify the target system and to accurately place it in its environment. It helps specify the issues at stake for the studied system along with the means it uses and the services it has to provide. At this stage, all the required information for risk management is

collected. This step involves three main activities, which are: definition of study scope, the preparation of the metrics, and the identification of the assets.

Phase 2 is called the feared events study. It contributes to the appreciation of risks by identifying and estimating the security needs for the primary assets in terms of confidentiality, integrity and availability. It consists also of identifying the impacts the threat sources if these security needs were not fulfilled.

Phase 3 involves studying threat scenarios that could cause the feared events by determining the threats affecting the supporting assets of the system. It identifies the attacks methods, the threat agents, the vulnerabilities, and the threats levels.

Phase 4 aims to estimate and assess the risks affecting the system and identify options to treat them. Finally, phase 5 aims to determine the security measures to be put into action and analyze the residual risks.

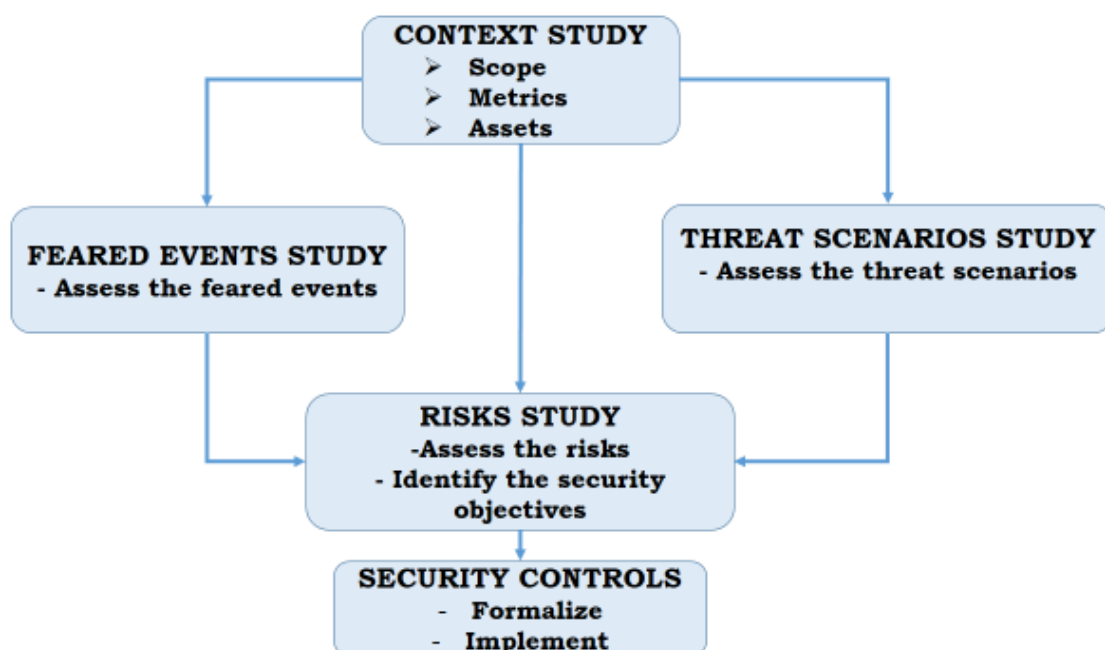


Fig.43 Phases of EBIOS method.

Further in this document, EBIOS is applied to Bitcoin system to identify and prioritize risks according to their importance and relevance in order to address the very urgent issues.

II.4 Bitcoin limitations and security issues

In the following section, we examine some Bitcoin limitations and security issues and we point out some mitigation measures that were suggested by the research community.

II.4.1 Bitcoin limitations

Despite its prominent success, Bitcoin is still suffering from some severe limitations and issues, such as throughput limits, the transaction confirmation latency, and the wastage of resources.

II.4.1.1 Throughput

The throughput, aka the Bitcoin scalability problem, refers to the limits in size and frequency that Bitcoin Blockchain network can deal with. Bitcoin can process up to 7 transactions per second. This is due to the limited size of the Block, which is 1MB; and the average block creation time, which is 10 minutes. This limitation may hinder the adoption of Bitcoin as a globalized payment system in the future. Currently Bitcoin cannot compete with other payment systems, such as VISA which recorded in 2015 a peak of 2000 transactions processed per second [67].

To overcome this serious limitation, multiple solutions were suggested, for instance, Bitcoin Forks mainly Bitcoin XT (became Bitcoin Cash in august 2017), and Bitcoin Classic [68]. These two Bitcoin forks increased the Block size and eventually increased the amount of transactions that can be processed per second.

II.4.1.2 Confirmation latency

In Bitcoin, the average block creation time is 10 minutes. This makes transactions confirmation take more than 10 minutes. This latency is due to the decentralized nature of the Bitcoin network and also to the security requirements. While it helps secure the system, it poses a serious issue that should be addressed to make fast payment services possible.

To dodge this limitations, many crypto-currencies clones of Bitcoin reduced the transaction confirmation time by decreasing the average time of Block creation. Litecoin reduces this latency to 2.5 minutes while Dogecoin made it almost 1 minute [69].

II.4.1.3 Wastage of energy

In Bitcoin ecosystem, mining is a resource-consuming process. Based on the Proof-of-work algorithm, mining wastes a huge amount of energy equivalent to \$15 million per day [70]. To avoid losing energy and money, miners join pools where they can pool their hashing capacity and share the reward among each other. Notwithstanding this alternative, Bitcoin community should consider other consensus mechanisms less resource-consuming such as the proof-of-stake.

II.4.2 Bitcoin security issues

Bitcoin has faced and is still facing many security challenges. The following paragraphs examines some security issues along with some suggested mitigation measures.

II.4.2.1 Zero-confirmation transactions security

In order for a new transaction to be spent, its owner should wait at least six confirmations, which takes about one hour or more. This constraint could hinder business using Bitcoin as a mean of payment. To overcome this limitation, Bitcoin urges its users to pay with their zero-confirmation

transactions, in which are transactions that have not yet been confirmed by miners. This alternative poses a security issue since it can be used to acquire services or goods without the buyers having to spend their bitcoins, which is a problem commonly known as the double-spending attacks.

Zero-confirmation transactions are insecure since attackers can easily mount double-spending attacks on them. This type of attacks has a probability of success of almost 100% when the attacker uses one or more helper node having as a role to disseminate his attack to a large number of connected nodes [71]. For this reason, zero-confirmations transactions should not be accepted directly by vendors.

One mitigation technique to counter this attack is that the vendor should consider a listening period of few seconds before delivering goods to the buyer. During this period, the vendor is more likely to detect the double-spending problem through monitoring of the network. However, the attacker is still able to find a way around the detection technique by delaying the transmission of the double-spending transactions in a way that exceeds the listening period, and also by increasing the number of helper nodes.

Another countermeasure that addresses the limitations of the listening period is to deploy observer nodes in the network to detect double-spending transactions. When five observers are deployed in different locations around the world, at least one observer detects double-spending attacks on zero-confirmation transactions [72]. In addition, Bitcoin system is currently punishing misbehaving nodes by banning them from connecting to other nodes for a period of time. Also, the system detects the double-spending problems through alert messages that are sent to the network by nodes which first detect the issue. Moreover, Bitcoin XT, which is a hard fork of Bitcoin that started in august 2017, has integrated a detection technique that would prevent double-spending attacks on zero-confirmations. This technique consists of peers forwarding the first double-spending transaction while dropping the others.

II.4.2.2 Blockchain forks

Blockchain security for Bitcoin and Ethereum is maintained by a proof-of-work consensus mechanism backed by miners who dedicate their computational power to create new blocks. When two blocks are found at the same time, the Blockchain is forked. This situation happens many times during the same day, but it is inherently resolved by the system as it considers the longest chain with the large difficulty as the valid version of the Blockchain. In case of forks, the transactions that do not appear in the valid version of the Blockchain are added later in subsequent blocks. When forks cannot be solved automatically by the system, Bitcoin developers can force chain at the expense of the other. This leverage held by developers question the decentralization of the Bitcoin system.

Bitcoin forks can be exploited to launch double-spending attacks more importantly on zero-confirmation transactions. Since eventually one chain is considered as the valid version of history, all the transactions that were included in the other version of the chain would be invalidated by miners. Some of these transactions will be included in subsequent blocks while the double-spent transactions will never be included. Bitcoin does not alleviate this problem by refunding the losing persons.

In 2013, Bitcoin experienced a severe fork when developers released Bitcoin client version 0.8 that implemented a LevelDB database instead of a BerkleyDB database that was used in version 0.7 [73]. This issue was solved by the intervention of Bitcoin developers who forced the smallest chain to be the valid version of Blockchain. This issue could have been avoided if Bitcoin developers had designed Bitcoin client 0.8 considering backward-compatibility.

The forking issue is very detrimental to the Bitcoin security and decentralization giving the leverage that Bitcoin developers hold in the system. The worst case scenario for Bitcoin is the collusion between Bitcoin developers and some miners to tamper with the Blockchain to serve their vested interest.

II.4.2.3 Transaction malleability

In Bitcoin, transactions are identified by the hash of their data. Whenever the data changes the transaction identifier changes too. In Bitcoin, transaction's signature, known also as the witness data, which unlocks the funds can still be valid despite some slight changes. This change in the witness data produces a new identifier of the same transaction. This vulnerability is known as the transaction malleability.

This issue was studied by *C. Decker and R. Wattenhofer* who examined allegations claiming that *MtGox* have lost 850,000 bitcoins due to malleability attacks. They concluded that barely 386 bitcoins could have been stolen using malleability attacks from *MtGox*. They also mentioned that transaction malleability is a real problem and should be addressed in any Bitcoin client implementation. The same issue was studied by *M.Andrychowicz, S. Dziembowski, D.Malinowski, and L. Mazurek* who suggested a deposit protocol with a timed commitment scheme to create a malleability-resilient "refund" transaction in order to prevent malleability.

This security issue was solved with the segregated witness protocol. This protocol defined a new structure known as the witness, which contains the scripts and signatures that redeem the funds. The witness is committed to blocks separately from the transaction Merkle tree [74].

II.4.2.4 Bitcoin majority attack

Bitcoin Blockchain security is strongly tied to an underlying assumption, which claims that the miners will always stay honest and work for the Bitcoin security and never against it since they are rewarded for this purpose through financial incentive. However if a malicious pool of miners holds more than 51% of the network computational power, this will make the network vulnerable to what is known as the 51% attack, the majority attack or selfish mining. *Beikverdi et al* claimed that Bitcoin is not decentralized since the mining market is hold by a few large mining pools and this issue increases the risk of a 51% attack [75]. They also claimed that in 2014 Bitcoin was only decentralized up 33%.

G. Karame et al mentioned that if an adversary holds more 50% of the computing power he can, in theory, double-spend transactions, prevent transactions from being confirmed, prevent honest miners from mining valid blocks in way that could invalidate the whole security of the network [76].

Ittay Eyal and Emin Sirer suggested that selfish miners who are detaining more than 33% of the network hashing power can still acquire an important part in the mining process. They also mentioned that a selfish mining strategy consists of a miner not announcing his mined blocks to the network in order to increase their revenue and letting other miners wasting their time and computational power [77].

To deter selfish mining *Ittay Eyal and Emin Sirer* suggested a strategy that urges miners to disseminate all the received blocks and to choose randomly one block to mine on it in case of two competing blocks [77].

II.4.2.5 Double-spending

Double spending happens when the same funds are spent twice. Bitcoin innovation came with the premise of solving this issue through a proof-of-work-based consensus. Despite this strong security control, double spending attacks are more likely to succeed in case of a Blockchain fork. Also, double-spending attacks on fast payments succeed with considerable probability and can be mounted at low cost [78]. *G. Karame et al* examined this issue for fast payments and suggested lightweight countermeasures that enable the detection of double-spending attacks. These measures have been implemented in most of Bitcoin forks.

One built-in way to prevent double-spending in Bitcoin is that the receiver of the funds cannot spent the received bitcoins until six blocks are added on the top of the block that contains the receiver's transaction. Another way is the transaction verification which consists of checking the signature, format, correctness of the fields, balance sufficiency, and the inputs were not spent in earlier transactions.

II.5. Bitcoin applications

Besides financial transactions, Bitcoin Blockchain serves many decentralized applications, for instance, decentralized storage, decentralized identity management, and even smart contracts.

II.5.1 Decentralized storage

The Blockchain technology allows any user to build a decentralized storage system which can ensure a high level of availability of the stored data. However, it is not recommended to store large data especially in the Bitcoin Blockchain.

To store data, the user generates a private key (K), then encrypts the data using the private key to produce an encrypted object (EO). He stores the encrypted object in n nodes and notes the URIs, Uniform Resource Identifier, (U1, U2, ...Un) of the stored EO in each node. After that, he encrypts all the URIs using the same key (K) to produce Encrypted URIs (EU1, EU2, ...EUn). Finally, he stores the Encrypted URIs (EUs: 1 to n) in the Blockchain along with a hash of the encrypted object (Hash(EO)) for a later integrity check. Once his transaction is confirmed, the user is then sure that the information is never altered. Fig. 44 depicts this process.

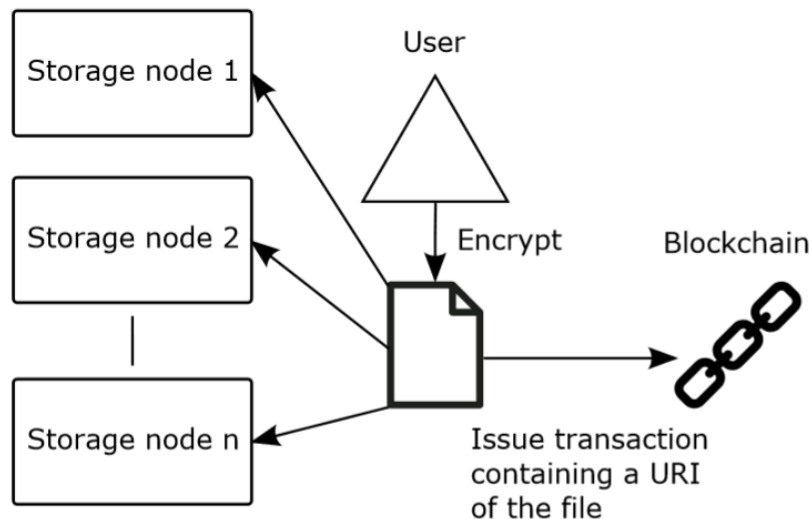


Fig.44 Decentralized storage in Bitcoin [79].

To retrieve the data stored in the Blockchain, the user retrieves the encrypted URIs (EUs: 1 to n), then he decrypts them using the private key (K). He uses the URIs to fetch the data stored in the n nodes. After that, he checks the integrity using the Hash (EO) retrieved from the Blockchain and a hash of the encrypted object stored in the nodes. Once the integrity is verified, he decrypts the encrypted object stored in the node using the private key to get the data.

II.5.2 Decentralized identity management

Blockchain also enables the construction of a decentralized identity management system by storing and confirming the identity in the ledger. This application denies to identity spoofing and help people keep their identity. Onecoin enabled this application using a dedicated Blockchain, known as Onecoin Blockchain [80].

II.5.3 Bitcoin smart contracts

Smart contracts are just like real-world contracts. They are in the form of tiny computer programs that can be stored in the Blockchain. These programs are executed once a certain goal or condition is reached between the two parties that create them.

Unlike Ethereum, which was designed with a built-in fully fledged Turing-complete programming language that can be used to create "smart contracts" [81], Bitcoin was implemented with a limited scripting language, which did not help enable smart contracts. However, using some new features added to Bitcoin through improvement proposals, certain smart contract functionality can be achieved through Bitcoin scripting. Bitcoin Improvement Protocol 65 (BIP65) introduced a new opcode, `OP_CHECKLOCKTIMEVERIFY`, which considered as the most important feature for smart contracts in Bitcoin. This opcode makes it possible to write scripts that prevent funds in a multi-signature wallet from being spent until a certain signature pattern is implemented or a certain amount of time passes [82].

Chapter III

Results

In this Chapter, we discuss the results of the work we accomplished in the research area. These results address the four research questions of this thesis.

III.1 Answer to the first research question

Our first paper entitled, ” *Risk Analysis Of Bitcoin Security Using EBIOS Method*”, is a qualitative risk analysis that provides an answer to the first question of this thesis, which is: “What are the major risks related to using Bitcoin as a crypto-currency and as a payment system?” Full paper is provided in appendix VI.

For this purpose, we applied the EBIOS method for Bitcoin in order to determine its major security risks. First, we determined the context of the study; then we identified the feared events and the threats scenarios followed by an analysis of the identified risks and finally we suggested some security measures in order to address these risks.

III.1.1 Context of the study

III.1.1.1 Scope of the study

This study aimed to assess information security risks pertaining to Bitcoin. It identified threat scenarios for the Bitcoin supporting assets, feared events for the primary assets, and threats and vulnerabilities. This study highlighted security measures designed to minimize the identified risks. It focused on the information security risks related to the Bitcoin system as a currency and as a payment system. We listed the main participants in the Bitcoin security as follows:

- Users and the stakeholders;
- Miners;
- Nodes in the Bitcoin peer-to-peer network
- Bitcoin’s community of developers.

We then identified the Bitcoin’s challenges as follows:

- Stay available for use around the clock 24/7;
- Include and validate all the valid transactions in the Blockchain every 10 minutes in average;
- Keep a clean and safe copy of the Blockchain in the majority of nodes;
- Solve continuously the fork-issue to avoid the double-spending problem.

This study concerned only the information security risks related to Bitcoin. It excluded the following risk areas:

- Social risks.
- Legal risks.
- Economic risks.

It examined the risks pertaining to the use of Bitcoin, mainly risks related to sending or receiving transactions, creating a new block and calculating its proof-of-work, propagating transactions and blocks to the connected nodes, storing a full and a clean copy of the Blockchain in the majority of the network’s nodes, and improving the Bitcoin protocol and upgrading it to continue addressing future needs. Fig.45 depicts in details the perimeter of the study.

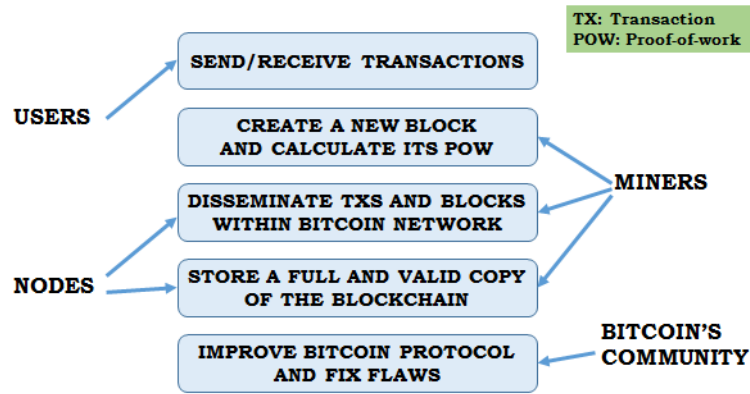


Fig.45 Perimeter of the study.

The plausible threat sources in the context of our study are as follows (see Table VIII):

TABLE VIII THREAT SOURCES

THREAT SOURCE TYPE	THREAT SOURCES
External human source, malevolent, with unlimited capabilities	State-sponsored attacks in the form of an APT
Internal human source, not malevolent, with weak capabilities	Imprudent user
External human source, malevolent, with significant capabilities	Hacker or group of highly skilled hackers competitors
Internal human source, not malevolent, with significant capabilities	Less serious administrator Less serious employee
External human source, malevolent, with weak capabilities	Cleaning personnel/Janitor Thief
Malevolent software of unknown origin	Flaw in the application Non-targeted virus
Natural phenomenon	Breakdown of material or network
Natural or health disaster	Earthquake/Fire/Flood/Tornado Illness or accident
Internal human source, malevolent, with unlimited capabilities	Greedy miner Malevolent administrator Subversive miners
Internal human source, malevolent, with significant capabilities	Malevolent Bitcoin developers Malevolent employees Corrupted nodes

III.1.1.2 Preparing the metrics

The preparation of the metrics aims to define a collection of parameters and scales that will serve to manage the risks related to Bitcoin, such as the security criteria and the scales of security needs. These criteria are factors that gauge the importance of different primary assets according to the business needs. The three unavoidable security criteria are defined as follow:

- Confidentiality: a property meaning that primary assets are accessible only to authorized personnel. In this context, the objective is to protect the identity of the Bitcoin user
- Integrity: a property of exactness and completeness of the primary assets. This means that primary assets are not altered;
- Availability: a property meaning that the primary assets are accessible at any giving time;

In this study we used the following scale levels (see TABLE IX) for the retained security criteria.

TABLE IX THE RETAINED SECURITY CRITERIA AND THEIR SCALE LEVELS

SECURITY CRITERIA	SCALE LEVEL	DETAILED DESCRIPTION
CONFIDENTIALITY	1. Public	The primary asset is public.
	2. Limited	The primary asset must only be accessible to staff and partners.
	3. Reserved	The primary asset must only be accessible to the (internal) staff involved
	4. Private	The primary asset must only be accessible to people who have been identified and who need to know
INTEGRITY	1. Detectable	The primary asset can be corrupted but the alteration can be identified.
	2. Controlled	The primary asset can be corrupted, if the alteration is identified and the integrity of the primary asset can be restored
	3. Has integrity	The primary asset must be rigorously uncorrupted.
AVAILABILITY	1. More than 48h	The primary asset can be unavailable for more than 48 hours.
	2. Between 24h and 48h	The primary asset must be available within 48 hours
	3. Between 4h and 24h	The primary asset must be available within 24 hours
	4. Less than 4h	The primary asset must be available within 4 hours

For risk assessment, we needed also to establish two scales: a scale of severity (gravity) and a scale of likelihood. A scale of severity describes all the possible levels of impact. TABLE X shows the scale levels retained for evaluation of severity.

TABLE X SCALE LEVELS OF SEVERITY

SCALE LEVEL	DETAILED DESCRIPTION
1. Negligible	Bitcoin will overcome the impact with no difficulty
2. Limited	Bitcoin will overcome the impact despite some difficulty
3. Important	Bitcoin will overcome the impact with serious difficulty
4. Critical	Bitcoin will not overcome the impact (its survival is threatened)

The scale of likelihood describes all the possible levels of likelihood of the threat scenarios. The following table (TABLE XI) illustrates these scale levels.

TABLE XI THE LIKELIHOOD SCALE LEVELS.

SCALE LEVEL	DETAILED DESCRIPTION
1. Minimal	This have not to recur
2. Significant	This could recur
3. Strong	This should recur
4. Maximal	This will certainly recur in the future

Prior to any risk analysis study, EBIOS requires the establishment of risk management criteria, which is a set of rules that help make decisions throughout the study. These criteria help estimate and evaluate the risks and also the way they should be addressed. TABLE XII shows some of the generic risk management criteria retained for this study.

TABLE XII RISK MANAGEMENT CRITERIA.

ACTION	RISK MANAGEMENT CRITERIA
2.1.1. Analysis of all the feared events	The feared events are estimated in terms of severity according to the defined scale of levels.
2.1.2. Assess each feared events	The feared events are ranked in a decreasing order of severity.
3.1.1. Analysis of all the threat scenarios	Threat scenarios are estimated in terms of their likelihood according to the defined scale of levels
3.1.2. Assess each threat scenario	The Threat scenarios are ranked in a decreasing order of their likelihood.

4.1.1. Analyze the risks	The severity of a risk is equal to the considered feared events. The likelihood of a risk is equal to the maximal likelihood of all the threat scenarios linked to the considered feared event.
4.1.2. Assess the risks	Risks of critical severity and those of important severity with strong or maximal likelihood are to be considered intolerable. Risks of important severity and significant likelihood, and those of limited severity and strong or maximal likelihood are to be considered as very significant. Risks of important severity and minimal likelihood or those of limited severity with a significant likelihood are to be considered significant.
4.2.1. Choose options for risk treatment	Intolerable, very significant, and significant risks have to be reduced to an acceptable level, transferred or avoided if this is possible. Negligible risks can be accepted.
5.1.1. Determine the controls	Security measures must be selected according to the context in order to minimize or eliminate the threat scenarios by fixing a vulnerability or by limiting the impact.

III.1.1.3 Identifying the assets

This step aims to identify the assets within the perimeter of the study. These are the primary and the supporting assets that the studied system is made of. The primary assets represent the informational assets or the immaterial assets that we want to protect. In other terms, this means the assets for which the non-respect of security criteria (Confidentiality, Integrity, and Availability) will put them in danger. TABLE XIII lists the major primary assets selected for this study.

The supporting assets are the technical or non-technical components of the studied system that support the primary assets. The study of these assets is important since they may hold some vulnerabilities that the threat sources might exploit to hurt the security of the primary assets. Table XIV illustrates the supporting assets that are considered in this study.

TABLE XIII THE STUDY'S MAJOR PRIMARY ASSETS.

PRIMARY ASSETS	DESCRIPTION
PRIVATE KEYS	Bitcoin private keys are generated randomly and from which public keys and addresses are derived. These keys are used in the digital signature required to spend transactions and therefore serve as a proof of ownership.
TRANSACTION	Transaction is the most important data structure in the Bitcoin system, which allows users to send or receive bitcoins. It consists of two main parts, which are the input and the output. The input of a transaction contains an unlocking script, which is mostly a digital signature and a public key proving ownership of the bitcoin.
BLOCK	The block is a data structure that holds a set of transactions that are included in the Blockchain. Its structure is made of a header and a collection of transactions. Blocks are created by miners through a process called mining in which a proof-of-work is calculated for each block.
BLOCKCHAIN	The Blockchain is a distributed database that stores all the transactions that ever happened in the Bitcoin system. These transactions are stored in different blocks that are linked to each other.
CONSENSUS MECHANISM	The consensus is achieved through the calculation of the proof of work of the new block, which is disseminated to the connected nodes for validation. Nodes validate the new block and add it to their current valid Blockchain.

TABLE XIV THE STUDY’S MAJOR SUPPORTING ASSETS.

SUPPORTING ASSET		TYPE	DESCRIPTION
MINING SITES		PREMISES	Miners rely on large sites from which they run their mining machines.
WALLET APPLICATIONS	DESKTOP WALLET	SOFTWARE	Desktop application is used to send and receive bitcoins. It helps also the user to keep track of their balance. Transactions generated by this application are signed and checked using an ECDSA algorithm based on a private and public keys.
	MOBILE WALLET		Mobile Wallets run on smart-phone or tablet operating systems, for instance, Apple iOS and Android. They run a lightweight node and are simple and easy to use.
	ONLINE WALLET SERVICES		They are web sites relying on a third party server that provides the same services as a stand-alone wallet. They allow the users to remotely manage their transactions.
INTERNET		NETWORKS	Bitcoin relies heavily on the internet. Without the internet no transaction can be transferred. (Not selected in this study).
BITCOIN PEER-TO-PEER NETWORK			Bitcoin rely on a peer-to-peer network that interconnects the nodes of participants.
HARDWARE WALLET		HARDWARE	Bitcoin Wallets that save the users’ private keys in a hardware device. They are considered the best way to securely store large amounts of bitcoin.
USER’S DEVICE			Bitcoin users rely on electronic devices to send or receive bitcoins, such as computers, cellphones, tablets, etc.
BITCOIN NODES’ MACHINES			The connected machines to the peer-to-peer network, which play different roles such as: network routing functions, Blockchain database, wallet services, and the mining.
MINING MACHINES			With high difficulty and competition, miners went from using CPU/GPU mining to ASIC mining. ASIC, Application-Specific Integrated Circuits (ASIC), are machines with hard-coded mining algorithm.
BITCOIN USER		PEOPLE	The focus is on the average person who is using Bitcoin.
PAPER WALLET		PAPER	They are Bitcoin keys printed in QR-code format in a paper.

EBIOS requires to establish the relationship between the primary and supporting assets so risks within the perimeter of the study can be compiled later in phase 4. Table XV illustrates this linkage.

TABLE XV THE RELATIONSHIP BETWEEN THE SUPPORTING AND THE PRIMARY ASSETS.

Support Asset	TRANSACTION	BLOCK	BLOCKCHAIN	CONSENSUS MECHANISM	PRIVATE KEYS
PRM - The mining sites	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
SW - Wallet applications	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
HW - User's device	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
NTW - Bitcoin peer-to-peer network	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
HW - Bitcoin participating nodes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
HW - Mining machines	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
PER - Bitcoin users	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
HW - Hardware Wallet	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
PAP - Paper Wallet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

After selecting the assets that will be considered in our study, we should carry out a census of the existing security measures for the supporting assets. These are technical or non-technical controls that can be categorized in three types, which are:

- Preventive controls are measures that protect vulnerabilities and make an attack ineffective or decrease its impact;
- Protective controls are measures that discover attacks and activate preventive or corrective controls;
- Recovery (Restoration) controls are measures that are often associated with business continuity and disaster recovery.

Table XVI illustrates some of the existing security measures for Bitcoin. It may not be complete but it provides the basics of security controls that are or should be implemented in the studied system.

TABLE XVI EXISTING SECURITY MEASURES

LABEL	ASSOCIATED SUPPORTING ASSET	CATEGORY OF THE MEASURE		
		PREVENTIVE	PROTECTIVE	RECOVERY
Security of the premises	MINING SITES	X	X	
Air-conditioning		X		
Fire-fighting devices		X		
Access control using password	WALLET APPLICATIONS	X	X	
Wallet backups				X
Anti-malware solutions	USER'S DEVICE	X	X	
Network service security	BITCOIN PEER-TO-PEER NETWORK	X	X	
Business continuity plan	MINING MACHINES			X
Anti-malware solutions		X	X	
Anti-malware solutions	BITCOIN NODES' MACHINES	X	X	

III.1.2 Study of the feared events

At this stage of EBIOS, we identify the generic scenarios that we wish to avoid within the perimeter of the study. The thought process is done at the functional level rather than the technical level, which means that the focus is on the feared events affecting the primary assets and not on those impacting the supporting assets.

III.1.2.1. Analyzing the feared events

In this step of the process, we identify the feared events, affecting the primary assets, for each security criterion. We then list the security needs for each primary asset, the impact in case of non-respect of security measures and the related threat sources along with a level of severity. The found feared events are related to the Confidentiality, Integrity, and Availability of the transaction, the block, the Blockchain, the consensus mechanism, and the private keys. An excerpt of the results is illustrated in Table XVII.

TABLE XVII EXCERPT OF THE FEARED EVENT LIST

PRIMARY ASSET	SECURITY CRITERIA	SECURITY REQUIREMENTS	THREATS SOURCES	IMPACTS	SEVERITY
TRANSACTION	Confidentiality	Limited	<ul style="list-style-type: none"> ▪ Imprudent user ▪ Malevolent Bitcoin developers ▪ Flaw in the application 	<ul style="list-style-type: none"> ▪ Reputational damage ▪ Putting someone in danger ▪ Loss of credibility with users ▪ Loss of anonymity 	Limited

III.1.2.2. Assessing the feared events

Assessing the feared events involves judging how important these events are within the perimeter of the study taking into consideration the established risk management criteria. For this purpose, the feared events are then prioritized according to their severity. This study identified twelve feared events as illustrated in table XVIII.

TABLE XVIII THE IDENTIFIED FEARED EVENTS

SEVERITY (GRAVITY)	FEARED EVENTS
CRITICAL	<ul style="list-style-type: none"> ▪ Block – Availability/ Integrity ▪ Consensus mechanism - Availability ▪ Private keys Confidentiality/Availability/ Integrity ▪ Transaction – Availability
IMPORTANT	<ul style="list-style-type: none"> ▪ Blockchain – Availability/ Integrity ▪ Consensus mechanism - Integrity ▪ Transaction - Integrity
LIMITED	<ul style="list-style-type: none"> ▪ Transaction - Confidentiality
NEGLIGIBLE	
NOT RETAINED	<ul style="list-style-type: none"> ▪ Block – Confidentiality ▪ Blockchain – Confidentiality ▪ Consensus mechanism - confidentiality

III.1.3. Study of threat scenarios

This step of the EBIOS method involves identifying the generic threat scenarios that could harm the information security of Bitcoin within the established perimeter of the study. These threat scenarios affect chiefly the supporting assets and not the primary assets. For this purpose, the thought process is carried out, at the technical level rather than the functional level.

III.1.3.1. Analyzing the threat scenarios

We identify the threat scenarios affecting each supporting asset and for each security criterion and then estimate their likelihood. We consider all the elements that participate in the threat scenarios such as the threats, the vulnerabilities, and the threat sources. An excerpt of the result of this study is shown in Table XIX.

TABLE XIX EXCERPT OF THE THREAT SCENARIOS LIST

S.A	SECURITY CRITERION	THREAT SOURCES	THREAT	VULNERABILITIES	PREREQUISITE	LIKELIHOOD
Paper Wallet	Confidentiality	<ul style="list-style-type: none"> ▪ Imprudent user ▪ Thief 	<ul style="list-style-type: none"> ▪ spying a paper wallet ▪ wear of a paper wallet ▪ Loss or theft of a wallet paper ▪ spying a paper wallet 	<ul style="list-style-type: none"> ▪ Allows the observing of interpretable data ▪ Poor quality constituents (fragile, easily flammable, subject to aging, etc.) ▪ Not suitable for the conditions of use (sensitive to humidity, etc.) ▪ Portable ▪ Poor quality constituents (fragile, easily flammable, subject to aging, etc.) ▪ Poor quality constituents (fragile, easily flammable, subject to aging, etc.) ▪ Not suitable for the conditions of use (sensitive to humidity, etc.) ▪ Not suitable for the conditions of use (sensitive to humidity, etc.) 	<ul style="list-style-type: none"> ▪ Knowledge of the existence and location of the paper media ▪ Physical access to the paper media (legitimate or illegitimate accessing, or bypassing) ▪ Knowledge of the existence and location of the paper media 	Maximal

III.1.3.2. Assessing each threat scenario

Assessing a threat scenario means judging its importance within the perimeter of the study while taking into consideration the established risk management criteria. The identified threat scenarios are ranked according to their likelihood as shown in table XX.

TABLE XX EVALUATION OF THREAT SCENARIOS

LIKELIHOOD LEVEL	THREAT SCENARIOS
MAXIMAL	Bitcoin users – availability Paper wallet - confidentiality
STRONG	Bitcoin nodes – availability Mining machines – availability Paper wallet – availability Users device – availability Users device – confidentiality Wallet applications - confidentiality
SIGNIFICANT	Bitcoin nodes – integrity Bitcoin peer-to-peer network – availability Bitcoin peer-to-peer network – integrity Bitcoin users – confidentiality Hardware wallet – availability Hardware wallet – confidentiality Hardware wallet – integrity Mining machines – integrity Wallet applications – availability Wallet applications – integrity
MINIMAL	Bitcoin users – integrity Paper wallet – integrity Users device – integrity
NOT RETAINED	Bitcoin nodes – confidentiality Bitcoin peer-to-peer network – confidentiality Mining machines – confidentiality

III.1.4 Study of risks

At this phase of EBIOS method, we highlight the real risks hovering around the perimeter of the study. For this purpose, we will assess the risks related to Bitcoin and then identify the security objectives, which determine the way to address these risks.

III.1.4.1. Analyzing and assessing the risks

Analyzing the risks involves identifying those risks affecting the perimeter of the study. We then determine their severity and their likelihood in two steps. In the first step, we do not take into consideration the existing controls while in the second step we take them into account.

Risk analysis implies linking the feared events (FE) and the threat scenarios (TS). EBIOS suggests two ways to establish this linkage, which are:

$$R (\text{Risk}) = 1FE + 1TS \quad (09)$$

$$R (\text{Risk}) = 1FE + TS1+ TS2+... + TS_n \quad (10)$$

The first formula suggests a risk for each feared event and each threat scenario. This formula leads to multiple risks, which can be cumbersome. The second formula calculate a risk for each feared event and a set of threat scenarios, which may be more pertinent since feared events directly impact the primary asset that we want to protect. For these reasons, we chose the second formula. After applying it, we ended up with 12 risks, as numerous as the feared events. Table XXI illustrates these risks along with their estimation without and with security measures (SM).

TABLE XXI LIST OF THE MOST RELEVANT RISKS PERTAINING TO BITCOIN SECURITY

RISK LABEL	THREAT SCENARIOS	ESTIMATION WITHOUT SM		ESTIMATION WITH SM	
		SEVERITY	LIKELIHOOD	SEVERITY	LIKELIHOOD
R0 – Risk related to Transaction Availability	<ul style="list-style-type: none"> ▪ Wallet applications – Availability ▪ Users device – Availability ▪ Bitcoin P2P Network – Availability ▪ Bitcoin nodes – Availability ▪ Mining Machines – Availability ▪ Bitcoin users – Availability 	Critical	Maximal	Limited	Significant
R1 – Risk related to Transaction Integrity	<ul style="list-style-type: none"> ▪ Wallet applications – Integrity ▪ Users device – Integrity ▪ Bitcoin P2P Network – Integrity ▪ Bitcoin nodes – Integrity ▪ Mining Machines – Integrity ▪ Bitcoin users – Integrity 	Important	Significant	Limited	Significant
R2 – Risk related to Transaction Confidentiality	<ul style="list-style-type: none"> ▪ Wallet applications – Confidentiality ▪ Users device – Confidentiality ▪ Bitcoin users – Confidentiality 	Limited	Strong	Limited	Minimal
R3 – Risk related to Block Availability	<ul style="list-style-type: none"> ▪ Bitcoin P2P Network – Availability 	Critical	Strong	Limited	Significant

	<ul style="list-style-type: none"> ▪ Bitcoin nodes Availability – ▪ Mining Machines Availability – 				
R4 – Risk related to Block Integrity	<ul style="list-style-type: none"> ▪ Bitcoin P2P Network Integrity – ▪ Bitcoin nodes Integrity – ▪ Mining Machines Integrity – 	Critical	Significant	Limited	Minimal
R5 – Risk related to Blockchain Availability	<ul style="list-style-type: none"> ▪ Wallet applications Availability – ▪ Users device Availability – ▪ Bitcoin P2P Network Availability – ▪ Bitcoin nodes Availability – ▪ Mining Machines Availability – ▪ Bitcoin users Availability – 	Important	Maximal	Important	Minimal
R6 – Risk related to Blockchain Integrity	<ul style="list-style-type: none"> ▪ Wallet applications Integrity – ▪ Users device Integrity – ▪ Bitcoin P2P Network Integrity – ▪ Bitcoin nodes Integrity – ▪ Mining Machines Integrity – ▪ Bitcoin users Integrity – 	Important	Significant	Important	Minimal
R7 – Risk related to Consensus mechanism Availability	<ul style="list-style-type: none"> ▪ Wallet applications Availability – ▪ Users device Availability – ▪ Bitcoin P2P Network Availability – ▪ Bitcoin nodes Availability – ▪ Mining Machines Availability – 	Critical	Strong	Important	Significant
R8 – Risk related to Consensus mechanism Integrity	<ul style="list-style-type: none"> ▪ Wallet applications Integrity – ▪ Users device Integrity – ▪ Bitcoin P2P Network Integrity – ▪ Bitcoin nodes Integrity – ▪ Mining Machines Integrity – 	Important	Significant	Limited	Minimal
R9 – Risk related to Private Keys Availability	<ul style="list-style-type: none"> ▪ Wallet applications Availability – ▪ Users device Availability – ▪ Bitcoin users Availability – ▪ Hardware wallet Availability – 	Critical	Maximal	Important	Minimal

	▪ Paper wallet- Availability				
R10 – Risk related to Private Keys Integrity	▪ Wallet applications – Integrity ▪ Users device – Integrity ▪ Bitcoin users – Integrity ▪ Hardware wallet – Integrity ▪ Paper wallet- Integrity	Critical	Significant	Limited	Minimal
R11 – Risk related to Private Keys Confidentiality	▪ Wallet applications – Confidentiality ▪ Users device – Confidentiality ▪ Bitcoin users – Confidentiality ▪ Hardware wallet – Confidentiality ▪ Paper wallet- Confidentiality	Critical	Maximal	Important	Minimal

Risk Assessment involves judging the importance of the risks according to the pre-established risk management criteria. Some of these risks can be omitted if they are deemed weak. Fig.46 illustrates the evaluation of risks after taking into consideration the security measures.

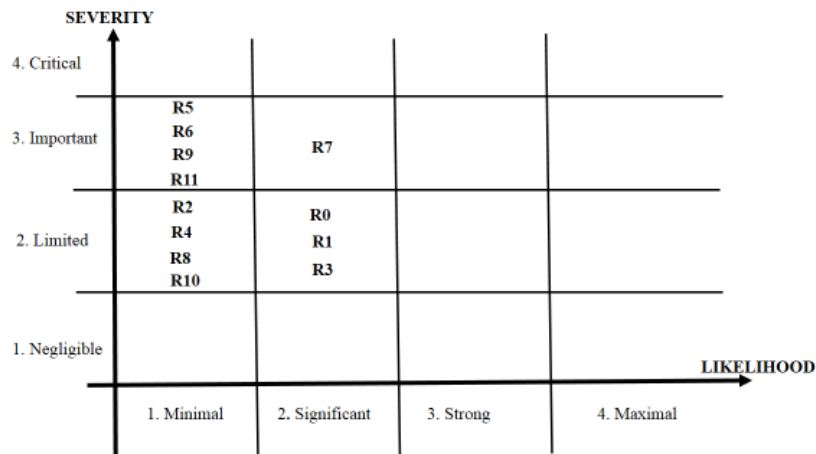


Fig.46. Risk assessment illustration

We assess the identified risks according to the risk management criteria established at the beginning of this study. For this purpose, we consider:

- **Intolerable Risks:** are those of critical severity and those of important severity with strong or maximal likelihood. In our case, we do not have any intolerable risks.
- **Very Significant Risks:** are those of important severity and significant likelihood, and those of limited severity and strong or maximal likelihood. In our case, R7 is a very significant risk.
- **Significant Risks:** are those of important severity and minimal likelihood or those of limited severity with a significant likelihood. In our case, these risks are: R0, R1, R3 and R5, R6, R9, R11.
- **Negligible Risks:** are those of limited severity and minimal likelihood. In our case, these risks are R2, R4, R8, and R10.

III.1.4.2. Identifying the security objectives

The analysis of risks pertaining to Bitcoin security showed four types of risks that should be addressed according to the established risk management criteria. At this stage, we should identify the options to treat these risks. There are four options to treat a risk, which are: Avoid (or refuse), Reduce, Accept, Transfer (or share).

According to the retained risk management criteria, intolerable, very significant, and significant risks have to be reduced to an acceptable level, transferred or avoided if this is possible. Negligible risks can be accepted. Therefore risks such as R0, R1, R3, R5, R6, R7 and R9 have to be reduced to an acceptable level while risks such as R2, R4, R8, and R10 can be accepted. R11 can be either reduced or transferred to a third party such as an insurance company to share the risk of loss or theft. Table XXII illustrates the security objectives for this study.

TABLE XXII SECURITY OBJECTIVES

RISK	SEVERITY	LIKELIHOOD	SECURITY OBJECTIVE
R0 – Risk related to Transaction Availability	Limited	Significant	Reduce
R1 – Risk related to Transaction Integrity	Limited	Significant	Reduce
R2 – Risk related to Transaction Confidentiality	Limited	Minimal	Accept
R3 – Risk related to Block Availability	Limited	Significant	Reduce
R4 – Risk related to Block Integrity	Limited	Minimal	Accept
R5 – Risk related to Blockchain Availability	Important	Minimal	Reduce
R6 – Risk related to Blockchain Integrity	Important	Minimal	Reduce
R7 – Risk related to Consensus mechanism Availability	Important	Significant	Reduce
R8 – Risk related to Consensus mechanism Integrity	Limited	Minimal	Accept
R9 – Risk related to Private Keys Availability	Important	Minimal	Reduce
R10 – Risk related to Private Keys Integrity	Limited	Minimal	Accept
R11 – Risk related to Private Keys Confidentiality	Important	Minimal	Reduce or transfer

After achieving every security objective, some of the risks might remain and should also be addressed. These are called residual risks that we must address with complementary security measures. For this purpose, we are going to consider the following rules:

- Avoided risks do not generate residual risks if they were completely avoided.
- Reduced risks lead to residual risks if they are not completely reduced
- Accepted risks are entirely residual risks
- Transferred risks do not imply residual risks.

These residual risks that will remain after the implementation of the security measures have also to be estimated in terms of severity and likelihood. Table XXIII illustrates the major residual risks along with their evaluation.

TABLE XXIII MAJOR RESIDUAL RISKS AND THEIR EVALUATION

RESIDUAL RISK	SEVERITY	LIKELIHOOD
Risk linked to the compromise of the Blockchain	Important	Minimal
Risk related to the disclosure of the Private Keys	Important	Minimal

In the next section, we determine the security measures that would lead to the achievement of the security objectives.

III.1.5. Security controls

At this stage of the EBIOS method, we determine the security measures that will allow us to achieve the security objectives, which means we need to highlight those controls that will allow us to avoid, reduce or transfer some of the identified risks.

The best way to mitigate risks is to adopt a strategy of defense in depth, which relies on three layers of defense. These layers are but not limited to: preventive layers, protective layers, and recovery layers

The following table presents a list of security measures destined to address the major risks related to Bitcoin in accordance with the security objectives.

Table XXIV SECURITY MEASURES DESTINED TO REDUCE OR TRANSFER THE RISKS RELATED TO BITCOIN.

SECURITY MEASURE	RISKS								ASSOCIATED SUPPORTING ASSET	CATEGORY OF THE MEASURE		
	R0	R1	R3	R5	R6	R7	R9	R11		PREVENTIVE	PROTECTIVE	RECOVERY
Wallet Backups Encryption	X	X			X	X	X	X	Wallet applications	X	X	
Monitoring and re-examination of third party service	X			X		X	X	X				

SECURITY MEASURE	RISKS								ASSOCIATED SUPPORTING ASSET	CATEGORY OF THE MEASURE		
	R0	R1	R3	R5	R6	R7	R9	R11		PREVENTIVE	PROTECTIVE	RECOVERY
Use wallets that implement mnemonic sentence (wallets that implement BIP-39)	X			X	X	X			Wallet applications Users device	X		X
Protection against exterior and environmental threat	X		X	X	X	X			The mining sites Bitcoin P2P network Mining machines	X		
Restrict visits in the Mining sites	X		X		X	X			The mining sites Mining machines	X		
Sensitive Assets inventory	X		X	X	X	X				X		
Management of removable media												
Make aware, qualification and training in matters of security	X		X	X	X	X	X	X	The mining sites Wallet applications (Personnel)	X		
Withdrawal of access right	X			X	X	X	X	X		X		
Use of anti-spying screens	X			X		X	X	X	User's device	X		
Choice of location and protection of hardware-use of anti-theft systems	X			X	X	X	X	X	User's device Hardware wallet	X		
Safe-guarding paper wallet from loss or theft							X	X	Paper wallet	X		
Measures against malevolent code- frequent update of anti-virus	X	X	X	X	X	X	X	X	Users device Mining machines Hardware wallet Bitcoin nodes	X	X	
Alternative power suppliers on case of power outage	X	X	X	X	X	X			Mining machines	X		
Sensitization about basics security measures	X	X		X	X		X	X	Bitcoin users	X		
Contracting an insurance to address private keys loss or theft							X	X	Paper wallet Wallet applications Hardware wallets	X		

This list of measures was mostly adapted from the ISO 27002 standard. These measures, if they are correctly implemented, would help decrease the severity and the likelihood of the most identified risks pertaining to Bitcoin. However, two residual risks may still remain and should be monitored so they cannot harm the functioning of the whole system. These residual risks are:

- Risk linked to the compromise of the Blockchain
- Risk related to the disclosure of the private keys

The underlying assumption of Bitcoin security was based on the honesty of the Miners. These miners gain more in staying honest through the reward they receive for new mined blocks and also through the transactions fees they collect. Although this assumption would stay strong for the future, Bitcoin stakeholders should invest in mining in order to keep the majority of the hashing power and therefore ensure more security for their bitcoins. This measure added to the frequent monitoring and risk assessment would reduce to an acceptable level the risk of the compromise of the Blockchain. Also, transferring the risk related to the disclosure of the private keys to an insurance company would definitely decrease its impact and by consequence hold this risk in a tolerable level. Furthermore, our second and third papers provide additional analysis and mitigation measures that address the risk of Blockchain compromise.

III.2 Answer to the second research question

Our second contribution entitled, “Bitcoin Difficulty, A Security Feature”, addresses the second question of this thesis, which is: “What is the relation Between the Bits, the Target, and the Difficulty; and How the Bitcoin difficulty contributes to the security of the system?” Full paper is provided in appendix VII.

Difficulty can be defined as a measure of how difficult it is to find a hash (proof-of-work) below a given target [83]. This parameter is set dynamically by the Bitcoin network every 2016 blocks or two-weeks in average. The difficulty is tied to two other parameters, the target and the Bits, which we will explain in the following paragraphs.

The proof-of-work serves as a proof that the miner has committed a great amount of hashing power to find the block header’s hash that satisfies the required condition. The proof-of-work is hard to find but easy to verify. It involves finding a value for the nonce that results in a block’s header hash, using SHA-256 algorithm, that is less or equal to the difficulty target (target). So how this target is calculated?

Every block contains a field called “Bits”, known also as target Bits, which is a four-byte number represented in a hexadecimal floating-point format. Bits value serves to calculate the difficulty target, which is used as a condition in the mining algorithm. The Bits field value of the first block in the Blockchain is 1d00ffff [84]. By convention, the first two digits (1d) represent the total number of digits a target is made of. It is used in the exponent of the floating-point notation while the remaining digits (00ffff) represents the coefficient. Now, how the target is derived from the Bits value?

To calculate the target from the Bits value, we rely on the following formula:

$$\text{TARGET} = \text{COEFFICIENT} * 2^{(8 * (\text{EXPONENT} - 3))} \quad (11)$$

Where:

- *COEFFICIENT* is the three Bytes on the right part of 4-Byte format of the Bits.
- *EXPONENT* is the first Byte on the left part of 4-Byte format of the Bits.

Using the hexadecimal representation and applying this formula to the block #0 with Bits value of (0x1d00ffff), the target would be:

$$\text{TARGET} = 0x00ffff * 2^{(0x8 * (0x1d - 0x3))} \quad (12)$$

Therefore, the result in hexadecimal format is:

$$\text{TARGET (in HEX)} = 0xffff00$$

We compare the header’s hash of the Block #0 (proof-of-work of Block #0) with the calculated target, using python 3. The following Script shows that the Block Header’s Hash is less or equal the calculated target, which means that the proof-of-work (POW) is valid.

```
>>> #calculating the target of Block #0 using the Bits Value
>>> Target = 0x00ffff*2**(0x8*(0x1d-0x3))
>>> # the decimal number is:
>>> print (Target)
26959535291011309493156476344723991336010898738574164086137773096960
>>> # the Target in hexadecimal representation
>>> hex(Target)
```

```
'0xffff000000000000000000000000000000000000000000000000000000000000'
>>> # Now let's compare this target to the block #0 header's hash
>>> BlockHash=0x000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
>>> BlockHash <= Target
True
```

The target condition sets the frequency at which a new proof-of-work is found. It determines also the difficulty for a collection of blocks. Since the computational power is increasing at a rapid speed and the Bitcoin network must keep the block generation time at 10 minutes in average, the target should adjust accordingly. The retargeting is happening dynamically on every full node independently for every 2016 blocks, which occurs every two weeks. The retargeting formula used by Bitcoin full nodes is [85]:

$$\text{NEW TARGET} = \text{CURRENT TARGET} * (\text{TIME ON MINUTES OF THE LAST 2016 BLOCKS}) / 20160 \text{ MINUTES.} \quad (13)$$

The difficulty is tightly linked to the target and shows how it is difficult to find a new hash of a block that satisfies the target condition. Its main purpose is to regulate the mining process, so a new block is mined every 10 minutes in average. It is calculated using the following formula [83]:

$$\text{DIFFICULTY} = \text{TARGETMAX} / \text{TARGETCURRENT} \quad (14).$$

Where:

- *TargetMax* is the target of the genesis block (Block#0)
- *TargetCurrent* is the target of the current block

The following script is used to calculate the difficulty of a Block using its target and the target of the genesis block. We used the block #495223, mined on Nov 20, 2017 10:53:40 AM, to verify this script.

```
#calculating the target of Block #0 as the Target Max using its Bits value
Target_Max = 0x00ffff*2** (0x8*(0x1d-0x3))
# the Max target in decimal number is:
print ("Max target Value in Dec(Block#0)=", Target_Max)
# the Max Target in hexadecimal representation
print("Max Target in Hex="+hex(Target_Max))
#calculating the target of Block #495223 using the Bits Value
Target_Current = 0x00ce4b*2** (0x8*(0x18-0x3))
# the current target in decimal number is:
print ("Current target Value in Dec(Block#495223)=", Target_Current)
# the Current Target in hexadecimal representation
print("Current Target in Hex(Block#495223)="+hex(Target_Current))
# Calculating the Difficulty
print("Difficulty=", round(Target_Max/Target_Current,2))
```

When running the script, we found the following results as depicted in Fig. 1. The calculated difficulty matches with the difficulty displayed in the Block #495223 Information (see Fig. 2).

```

Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/lam/Desktop/Conferences/EMENA-ISTL 2018/script1.py ====
Max target Value in Dec(Block#0)= 26959535291011309493156476344723991336010898738574164086137773096960
Max Target in Hex=0xffff00000000000000000000000000000000000000000000000000000000000000000000000000000000
Current target Value in Dec(Block#495223)= 19758940920085072387393228723348383373068660102939017216
Current Target in Hex(Block#495223)=0xce4b000000000000000000000000000000000000000000000000000000000000
Difficulty= 1364422081125.15

```

Fig. 47 Difficulty calculation of block #495223.

Difficulty	1364422081125.1474
Bits	1800ce4b
Size (bytes)	962992
Version	536870912
Nonce	2561881396
Next Block	495224

Fig. 48 Block #495223 information [86].

The difficulty is tightly linked to the hashing rate. When the hashing rate increases, the proof-of-work is found quickly and therefore the difficulty increases too to keep the proof-of-work finding around 10-minutes in average. Also, when proof-of-work discovery time is slower, the difficulty decreases. Table XXV illustrates the strong correlation between the difficulty and the hashing rate. It shows also that difficulty and the hash rates have quintupled since the last year. This is due mainly to competition between miners.

Table XXV. Difficulty and hash rate change between 2016 and 2017[87]

Date	Difficulty	Hash Rate (GH/s)
Dec 6 th , 2017	1,590,896,927,258	11,388,083,790
Dec 2 nd , 2016	286,765,766,821	2,052,749,317
Ratio of change	5.547722606133257	5.547722605818799

Without the difficulty, any miner possessing big hashing power would take over the Blockchain and could change it at will, therefore the difficulty participates strongly to the security of the Bitcoin.

All the aforementioned concepts suggest that Bitcoin is a secure by design crypto-currency. Its security relies on the cutting-edge cryptographic technologies such as the digital signature and the hash functions. The Difficulty plays a major role in the Bitcoin Security since it regulates the mining process, so a new block is added to the Blockchain within 10 minutes in average. Also, its dynamic change helps keep up with the increasing hashing rate to avoid Blockchain hijacking by miners with huge computational power. Notwithstanding the difficulty benefits, there is a big issue that Bitcoin community should address, which is the selfish mining. Our third contribution provides a new way to democratize the Bitcoin mining process.

III.3 Answer to the third research question

Our third paper entitled, “Using the Randomized Solution of the Dining Philosophers Problem to Prevent the Bitcoin Majority Attack”, provides an answer to the third question, which is: “How can we deter the majority attacks or the selfish mining?” Full paper is provided in appendix VIII.

III.3.1 Bitcoin majority attack

The majority attack, AKA the 51% attack, refers to the ability of a miner or a pool of miners controlling more than half of the network hashing rate [88]. This would grant them on the ability to generate the longest chain in the system. Basically, any malevolent mining pool with more than 51% of the network hashing rate, would be capable of:

- Modifying the transaction data, which may cause double-spending attack [89,90]
- Preventing confirmations [91]
- Preventing bitcoin generation [88]

Andreas M. Antonopoulos, an author and a Bitcoin advocate, said that if a state-sponsored 51% attack occurs, it will cost a huge amount of money to the attacker and may cause double-spending only in one block since this type of attack can be revealed by the Bitcoin community. The system will be split into two forks: one of the attackers and the other of the honest bitcoin community and few if none of the Bitcoin community would join his fork [92]. This statement proves that the 51% attack is still possible despite its high cost and its low probability of occurrence.

As mentioned above in this paper, the 51% attack was studied by other researchers who suggested different solutions. Besides these solutions, Bitcoin community suggests to always monitor the mining pools hashing rate and try to balance them, so no one will gain more than 50% of the hashing power, which seems to be a cumbersome control measure and not easy to be carried out. To add more insights about this issue, we are suggesting an out of the box solution based on the randomized solution of the dining philosophers’ problem.

III.3.2 Dining philosophers’ problem

The dining philosophers’ problem (DPP) is a classic synchronization problem which is used to evaluate situations where there is a need of allocating multiple resources to multiple processes [93].

There are N numbers of philosophers sitting around a round table eating noodles and sharing ideas as well as thinking. Each philosopher requires two chopsticks to eat, and there is one chopstick between two philosophers. The purpose is that no one starves, and maximum number of philosophers can eat at the same point of time [94]. At any moment, a philosopher is either eating or thinking.

In the dining philosophers’ problem, two neighbors cannot eat at the same time. The maximum number of philosophers eating at the same time is equal to the integer part of $N/2$, where N is the number of philosophers. There are three states for each philosopher: hungry, eating, and thinking. Fig. 49 shows 8 dining philosophers sitting around a table.

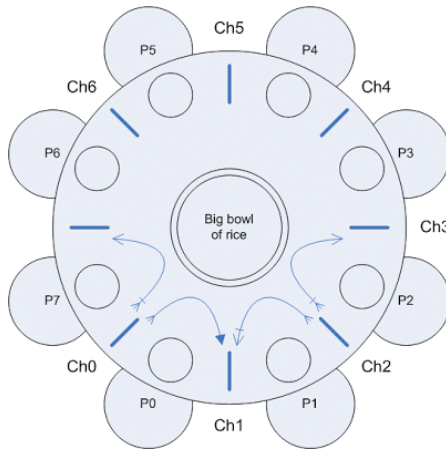


Fig. 49. Eight sitting dining philosophers [95]

A solution to this dining philosopher problem is represented in Fig. 50:

```
void Philosopher (int i)
{
    while(True)
    { THINK;
      PICKUP(CHOPSTICK[i]);
      PICKUP(CHOPSTICK[i+1 mod N]);
      EAT;
      PUTDOWN(CHOPSTICK[i+1 mod N]);
      PUTDOWN(CHOPSTICK[i]);
    }
}
```

Fig.50. A basic solution of the DPP

This code solves the DPP, but it could lead to a deadlock, in which no philosophers is able to eat because one of his required chopstick would be taken by another philosopher. This happens when philosophers take one chopstick at the same time, only the left or only the right, as shown in the Fig. 51.

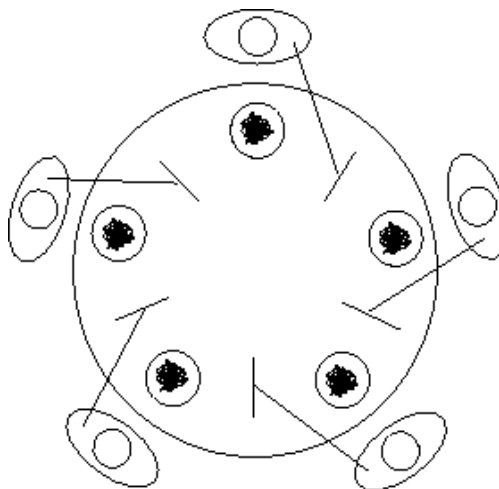


Fig.51. A deadlock in the DPP [96].

The easiest way to avoid the deadlock is to make sure that at least one philosopher is taking his first chopstick from the right side while the others are taking their first chopstick from the left side or vice versa. Fig. 52 illustrates this solution [97]:

```
void Philosopher(int i)
{
```



```

while(True)
{ THINK;
  PICKUP(CHOPSTICK[min(i, i+1 mod N)]);
  PICKUP(CHOPSTICK[max(i, i+1 mod N)]);
  EAT;
  PUTDOWN(CHOPSTICK[min(i, i+1 mod N)]);
  PUTDOWN(CHOPSTICK[max(i, i+1 mod N)]);
}
}

```

Fig.52. A solution to the deadlock problem in the DPP

Another solution to the deadlock problem is to introduce a central entity, as an arbitrator, who gives permission to only one philosopher at a time and make sure that the philosopher has picked up both of his chopsticks. This solution, though efficient, adds a little of centralization in the system. Other solutions to the same problem are available, such as Chandy/Misra solution [98] and Dijkstra's solution [99].

Besides the deadlock, another problem, known as starvation may occur. This happens when one or more philosophers will not be able to eat at all because other philosophers may monopolize the chopsticks. Fig. 53 depicts this issue, where Philosophers A&C, B&E can take turns to pick the chopsticks to such a degree that philosopher D starves out.

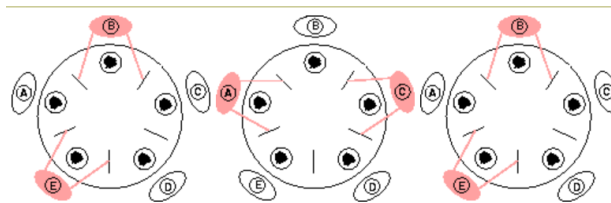


Fig.53. Starvation issue in the DPP [93].

A good solution of the DPP should not have any deadlock or starvation. In this context, Lehmann and Rabin solved the resource-starvation problem in the following way:

A philosopher will not pick up his/her neighbor's chopstick (when it has no chopsticks) if that neighbor is trying to eat and has not eaten since the philosopher's most recent meal [100].

In the next section, we will rely on this implementation to provide an alternative way to prevent the 51% attack in Bitcoin.

III.3.3. Contribution

In this research paper, we suggested the randomized solution of the dining philosophers' problem to prevent the Bitcoin 51% attack. Before being able to race for a new POW, mining pools should get their chopsticks through an arbitrator node, which should organize this process to avoid the deadlock or the starvation problems. Fig. 54 illustrates this process for five mining pool.

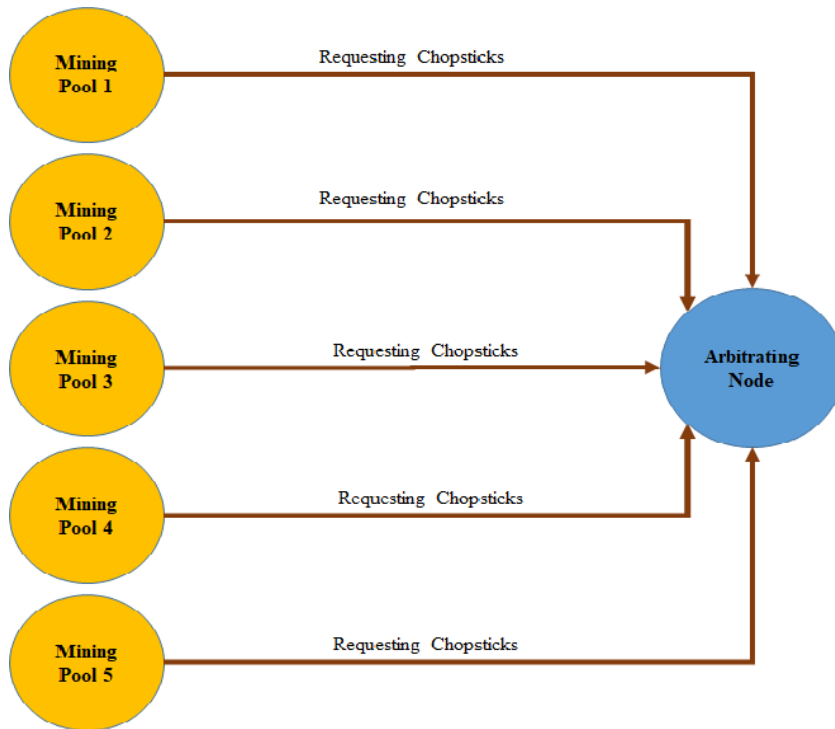


Fig.54. Arbitrating node in Bitcoin

The arbitrating node that we are suggesting will make sure that every mining pool will get a chance to mine and no one will monopolize the mining process. The mining process will be organized in two phases: picking the chopsticks and mining. While the first phase adds some centralization in the system, the second one will still be completely decentralized.

For this purpose, we adapted the Lehmann and Rabin solution to the mining process by:

- Replacing Philosophers with mining pools.
- Replacing the three states of philosophers (Hungry, Eating, and Thinking) with three states for miners (Hungry to mine, Mining, resting to cool down equipment and stop consuming electricity).

Picking two chopsticks (right and left) is a required condition to start the mining race. This condition will deny to any mining pool, regardless of its hashing capacity, a hijacking of the mining process and will give all mining pool a fair chance to win a reward in the system.

Currently, there are 19 mining pools in the system. With this implementation, only 9 pools (the integer part of $19/2$) will be competing at a time for a POW.

This paper relies on Markov Decision Process (MDP), to simulate the dining philosophers' problem solution based on the Lehmann and Rabin's randomized solution. MDP is widely used by the research community, which allows us to take advantage of previous works and the available literature.

To build the model, we relied on a model checker software called Prism. Prism provides analysis for systems that shows a probabilistic behavior. It is used for different research areas, such as communication and multimedia protocols, stochastic algorithms, security protocols, and biological systems. PRISM can build and analyze several types of probabilistic models [101]:

- discrete-time Markov chains (DTMCs)
- continuous-time Markov chains (CTMCs)
- Markov decision processes (MDPs)
- probabilistic automata (PAs)
- probabilistic timed automata (PTAs)

In this study, we are using Markov decision process to analyze the solution we are suggesting for the 51% attack. The following diagram (see Fig. 55) illustrates the 12 states of the process and the actions taken between two states.

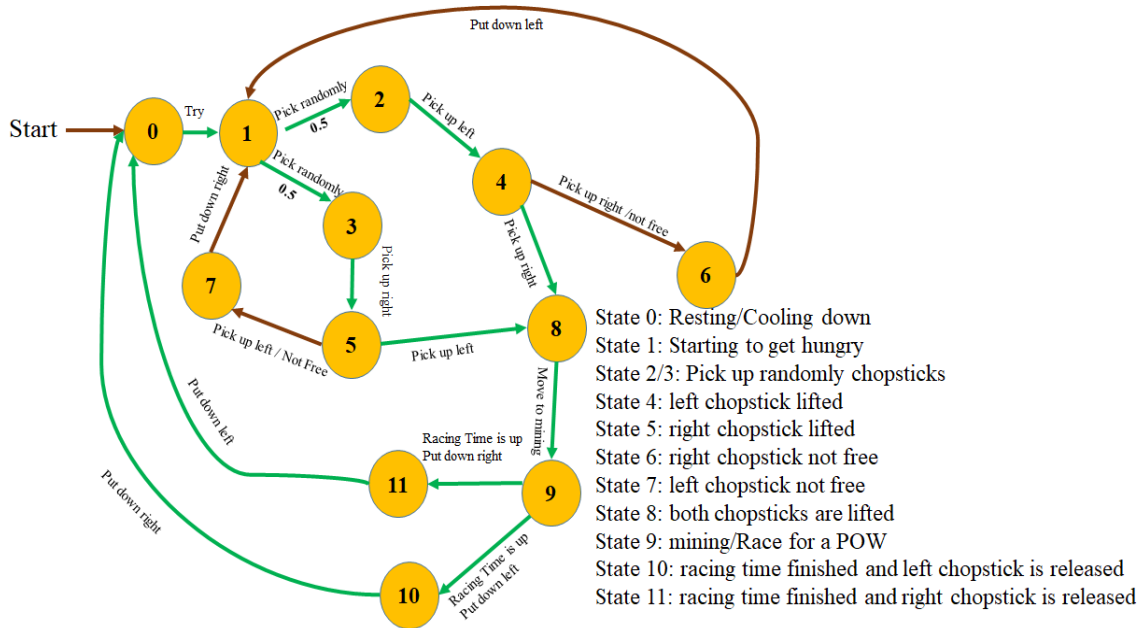


Fig. 55. State diagram of the process.

We split the three states of a mining pool into twelve states to see specific details of the whole process. This model is implemented in prism for three miners as illustrated in the following code as shown in Fig. 56 [102]:

```
// randomized dining philosophers adapted to Bitcoin Mining Process
// left Chopstick free and right Chopstick free resp.
// left neighbor is Miner 2
formula lfree = (M2>=0&M2<=4)|M2=6|M2=10;
// right neighbor is Miner 3
formula rfree = (M3>=0&M3<=3)|M3=5|M3=7|M3=11;
module Miner1
  //12 state-diagram
  M1: [0..11];
  [] M1=0 -> (M1'=1); // trying
  [] M1=1 -> 0.5 : (M1'=2) + 0.5 : (M1'=3); // pick randomly
  [] M1=2 & lfree -> (M1'=4); // pick up left chopstick
  [] M1=3 & rfree -> (M1'=5); // pick up right chopstick
  [] M1=4 & rfree -> (M1'=8); // pick up right chopstick (got left)
  [] M1=4 & !rfree -> (M1'=6); // right chopstick not free (got left)
  [] M1=5 & lfree -> (M1'=8); // pick up left chopstick (got right)
  [] M1=5 & !lfree -> (M1'=7); // left chopstick not free (got right)
  [] M1=6 -> (M1'=1); // put down left chopstick
  [] M1=7 -> (M1'=1); // put down right chopstick
  [] M1=8 -> (M1'=9); // move to Mining (got both chopsticks)
  [] M1=9 -> (M1'=10); // racing time finished and left chopstick is released
```

```

[] M1=9 -> (M1'=11); // racing time finished and right chopstick is released
[] M1=10 -> (M1'=0); // put down right chopstick and return to resting
[] M1=11 -> (M1'=0); // put down left chopstick and return to resting
endmodule
// construct further modules through renaming
module Miner2 = Miner1 [ M1=M2, M2=M3, M3=M1 ] endmodule
module Miner3 = Miner1 [ M1=M3, M2=M1, M3=M2 ] endmodule
// rewards (number of steps)
rewards "num_steps"
[] true : 1;
endrewards
// labels
label "Hungry" = ((M1>0)&(M1<8))&((M2>0)&(M2<8))&((M3>0)&(M3<8));
label "Mining" = ((M1>=8)&(M1<=9))&((M2>=8)&(M2<=9))&((M3>=8)&(M3<=9));

```

Fig. 56. Randomized solution of the DPP adapted to Bitcoin mining process

We relied on the following model checking, which is depicted in Fig. 57, to check the number of iteration and the time it takes for each miner get a chance to race for a POW; we checked also the number of iterations before the first miner get a chance to mine.

```

const int K; // discrete time bound
// liveness (if a Miner is hungry then eventually some Miners race for a POW)
"Hungry" => P>=1 [ true U "Mining" ]
// bounded waiting (minimum probability, from a state where someone is hungry, that a Miner will mine within K steps)
Pmin=?[true U<=K "Mining" {"Hungry"}]{min}
// expected time (from a state where someone is hungry the maximum expected number of steps until a Miner mines)
Rmax=?[F "Mining" {"Hungry"}]{max}

```

Fig.57. Model checking.

The results below illustrate some statistics for the MDPs we have built for different values of the constants N=3 (number of miners) and K (number of iterations).

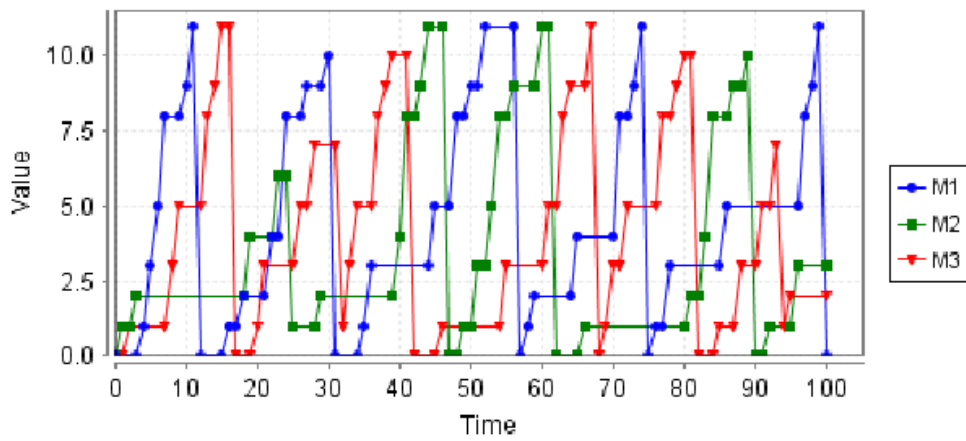


Fig. 58. Miners' different states (0 to 11) for 100 iterations

Fig.58 shows that miner 1 start mining after 10 iterations, while miner 2 and 3 got their chances after 14 and 44 iterations respectively. Fig.59 depicts that the minimum time required for a miner to start mining is around 0.0625 seconds and the maximum iterations needed for all miners to get a chance to mine is about 50.

```

Property:
  "Hungry"=>P>=1 [ true U "Mining" ]

Defined constants:
  <none>

Method:
  Verification

Result:
  true (property satisfied in the initial state)

```

Property Details

```

Property:
  Pmin=? [ true U<=K "Mining" {"Hungry"}{min} ]

Defined constants:
  <none>

Method:
  Verification

Result:
  0.0625 (minimum value over states satisfying filter)

```

Property Details

```

Property:
  Rmax=? [ F "Mining" {"Hungry"}{max} ]

Defined constants:
  <none>

Method:
  Verification

Result:
  50.99852574264277 (maximum value over states satisfying filter)

```

Fig.59. Results of the model checking properties.

The results change as we change the number of miners. For more analysis and details, we tested one implementation of the randomized solution of the dining philosophers adapted to the mining process. The following paragraphs provide more detail.

For testing purposes, we used 19 mining pools, numbered from 1 to 19 and an adaptation to a solution of the DPP, written in Java language, to simulate the role of the Arbitrating Node [103]. Fig. 60 shows an excerpt of the used code and some of the results.

```

while (true) {
    // Resting
    doAction(System.nanoTime() + ": Resting");
    synchronized (leftChopstickk) {
        doAction(
            System.nanoTime()
            + ": Picked up left ChopStick");
        synchronized (rightChopstickk) {
            // eating
            doAction(
                System.nanoTime()
                + ": Picked up right Chopstick - Mining");

            doAction(
                System.nanoTime()
                + ": Put down right Chopstick");
        }
    }
}

```

Code

```

Mining Pool:8 309921003803648: Resting
Mining Pool:7 309921003843549: Picked up right Chopstick - Mining
Mining Pool:15 309921034319353: Put down left Chopstick. Back to Resting
Mining Pool:16 309921034334760: Picked up left ChopStick
Mining Pool:13 309921044387485: Resting
Mining Pool:12 309921044392226: Picked up right Chopstick - Mining
Mining Pool:7 309921054014729: Put down right Chopstick
Mining Pool:18 309921066046710: Resting
Mining Pool:17 309921066089377: Picked up right Chopstick - Mining

```

Results

Fig. 60. Excerpt of code and results used for testing

After running this code several times, we realized that all the mining pools got a chance to mine and eventually add a new block in the blockchain independently of their hashing rate (see Fig. 61). The results show that mining pools get their chopsticks randomly. This denies to miners the ability to mine many blocks successively, which is a condition that is difficult to reverse in Bitcoin.

14 16 8 18 11 6 13 10 15 17 18 5 12 9 14 16 4 13 11 15 8 17 10 3 14 16 18 9 12 7 2 17 11 8 13 1 6
18 10 16 12 9 19 5 7 15

Fig. 61. Randomized order of mining for the 19 mining pools different for each attempt.

Giving all these results, we concluded that the randomized solution of the DPP implemented within an arbitrating node would prevent the monopoly of the mining process for any super-pool, holding a hashing rate capacity of more than 50%.

Bitcoin is a secure by design crypto-currency that relies on cutting-edge cryptographic technologies such as the digital signature and the hash functions. In addition, the Difficulty plays a major role in the Bitcoin Security since it regulates the mining process, so a new block is added to the Blockchain within 10 minutes in average. Despite all these features, Bitcoin is still vulnerable to 51% attacks. This paper provided a new way to regulate the mining process, so no pool will hijack the system by using the randomized solution of the dining philosophers' problem, which should be implemented in an arbitrating node. This study provided also some analysis of the different states of mining pools using a Markov Decision Process model, implemented in Prism. The results showed that the suggested solution works perfectly to organize the mining process and grant each mining pool a chance in the POW race and prevent monopoly of super-pools. Therefore, we are strongly recommending Bitcoin community to consider this alternative as a way to prevent the 51% attack. Bitcoin community should consider this solution while working on ways to make it decentralized or yield for some centralization for security purposes. The following research paper provides an analysis to determine the center of gravity of Bitcoin for a better defense.

III.4 Answer to the fourth research question

Our fourth contribution entitled, “Bitcoin Embedded Security Items Review and Center of Gravity Analysis”, addresses the fourth question which is: “What are the Bitcoin embedded security items and what is the Bitcoin center of gravity; and what should be done to disrupt or secure the system?” Full paper is provided in appendix IX.

III.4.1. Bitcoin embedded security features

In this section, we will go through the most important security features that are embedded within the Bitcoin system. These features are brought by the cutting-edge innovations in the realms of cryptography and distributed systems. They include hashing functions, elliptic curve, digital signature, and encodings. The analysis is done based on three security objectives, which are Confidentiality, Integrity, and Availability, also known as CIA.

III.4.1.1 Bitcoin keys and addresses security

A Private Key, which serves as a proof-of-ownership, is a 256-bit number picked randomly using the operating system entropy (randomness). It is derived from random function or a hash of some data. There are 2²⁵⁶ numbers of possibilities, which makes it hard to predict. In addition, private keys are stored in an encrypted format using a symmetric encryption system, which is the advanced encryption system (AES).

TABLE XXVI: PRIVATE KEY SECURITY IN TERMS OF C.I.A

	Confidentiality	Integrity	Availability
Private Key	- Depends on the level of randomness - Tied also to the AES encryption System including the key management.	- Ensured by the WIF encoding - The AES encryption system.	Tied to the availability of the Bitcoin wallet

A Public Key, which represents a point in the Elliptic Curve, defined by secp256k1 Standard [104], is related to the private key. It is derived using a discrete logarithm formula (See Equation 15), is proven to be hard to break giving the current computational power. Public key security relies on the unbreakable the elliptic curve discrete logarithm (ECDL) formula, where G is the Generator point of the Koblitz Elliptic Curve (Equation 16)

$$PUBKEY = PRIVKEY * G \quad (15)$$

$$EC: Y^2 = X^3 + 7 \quad (16)$$

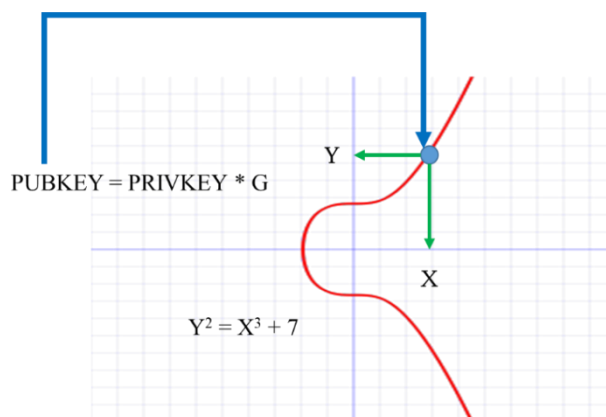


Fig.62. Secp256k1 defined elliptic curve.

TABLE XXVII. PUBLIC KEY SECURITY IN TERMS OF C.I.A

	Confidentiality	Integrity	Availability
Public Key	Relies on the discreet logarithm formula and the no-backdoor in the elliptic curve until now	Ensured by Base58Check encoding	Tied to the availability of the Bitcoin wallet

Bitcoin Address, identifies users and serves to send and receive funds from one user to another. It is derived from the public key using hash-functions, such as SHA-256 and RipeMD-160 (See Equation 17).

$$\text{ADDRESS} = \text{RIPEMD-160}(\text{SHA-256}(\text{PUBKEY})) \quad (17)$$

TABLE XXVIII. BITCOIN ADDRESS SECURITY IN TERMS OF C.I.A

	Confidentiality	Integrity	Availability
Bitcoin Address	Relies on the no-collision in the hashing functions: SHA-256 and RIPEMD160	Ensured by Base58Check encoding	Tied to the availability of the Bitcoin wallet

Overall, Bitcoin keys and addresses security depends on:

- The randomness used in producing the private key
- The elliptic curve discrete logarithm, which can't be solved giving the current computational power
- The one-wayness property of the hashing functions SHA-256 and RACE MD (RIPEMD-160)
- And also because no backdoors were discovered in the elliptic curve till now

If a collision is to happen, in the future, in SHA-256 function, Bitcoin community should be prepared to shift to SHA-512 in order to prevent producing the same address for different private keys.

IV.4.1.2 Bitcoin wallet security

Wallets are applications used to send/receive transactions, to track user's balances, and to store user's private/public keys. They run on different platforms, such as windows, Linux, MacOS, etc. They can be also in a hardware and paper format. Hardware wallets use electronic devices that store private keys and are considered the most secure of all kinds of wallets. Wallets security depends more on the user's awareness of the threats and risks that are related to private keys security, such as encryption. Bitcoin keys are encrypted, with a master key which is entirely random, using the Advanced Encryption Algorithm (AES) [105].

IV.4.1.3 Bitcoin scripting language

Bitcoin uses a stack-based language with simple and limited functions, known as opcodes. It supports functions that serve for comparison, hashing, and signature verification. These functions are mainly used to lock and unlock transactions. For security purposes, loop functions are disabled, which deny to any attacker the possibility of crafting denial of service attacks [106].

IV.4.1.4 Bitcoin transaction security

Bitcoin Transactions consist of inputs and outputs that define the sender and the receiver of the funds. They are secured using the Elliptic Curve Digital Signature (ECDSA) to ensure that funds can only be spent by their rightful owners [107]. This signature lies heavily on the elliptic curve security. If a backdoor is found in the secp256k1 elliptic curve, this signature will be vulnerable to brute force attacks. Therefore, Bitcoin transactions are secure as far as the elliptic curve digital signature is still secure.

IV.4.1.5 Bitcoin blocks security

Blocks are created by miners who succeed in finding the proper proof of work (POW). Their structure is based on a header and a set of valid transactions. The header contains a hash of the previous block, which plays a major role in the block integrity. Blocks are broadcast publicly to the connected nodes in the network. The integrity of the enclosed transactions is ensured by the Merkle tree, which provides a hash of the included transactions.

IV.4.1.6 Bitcoin's Blockchain security

In Bitcoin, each block is linked to the previous one. These chain of blocks is what makes the Blockchain. It is also made public in the network. The back-linkage of blocks helps ensure some security for the Blockchain. If a block is altered, all the following blocks should be altered and their POW puzzle should be resolved, which is time and resources consuming process. It is computationally impossible to change a block after it has been confirmed by six other blocks [108]. Sometimes, a fork in Blockchain happens when two blocks are found at the same time by two different miners. Bitcoin solve this issue by considering the longest chain with the largest difficulty as the valid version of history.

IV.4.1.7 Mining process security

Miners get reward for each mined block. This incentive works to keep miners working for the system and not against it. The reward is currently 12.5 BTC and this help them offset the cost of consumed electricity.

The difficulty regulates the block production process, so a new block is added to the Blockchain every 10 minutes in average. This feature helps Bitcoin system to adapt to the continuously increasing hashing rate.

III.4.2 Center of gravity analysis

In this section, we define the center of gravity for any given system and we provide a quick analysis of critical factors related to Bitcoin.

III.4.2.1 Definitions

According to US Army JP5-0, a COG is a source of power that provides moral or physical strength, freedom of action and the will to act. A COG is analyzed within a framework of three critical factors, which are: Critical capabilities (CC), Critical requirements (CR), and Critical vulnerabilities (CV) [109].

- CC is the primary ability, it is what the COG is able to do ;
- CR is essential conditions, resources, or means required by which the COG performs its CC ;
- CV are CRs that are deficient or vulnerable; they may be transient and internal or external.

III.4.2.2 Critical factors analysis

The following table provides an overview of the conducted analysis of the critical factors and a suggestion of a center of gravity for Bitcoin at two different levels: the strategic one which depicts high-level goals, and the operational one which represents the daily, weekly, monthly actions that should be done for the survival of the system.

TABLE XXIX: AN OVERVIEW OF THE BITCOIN CRITICAL FACTORS ANALYSIS

<p>COG:</p> <ul style="list-style-type: none"> • Bitcoin Strategic COG : is the trust that has gained within its users ; • Bitcoin operational COG: the consensus mechanism (POW) that allow the system to confirm TXs, Create new Block, and generate brand new bitcoins. 	<p>CC:</p> <ul style="list-style-type: none"> • create new units of currency (bitcoins created through the mining process) • send/receive bitcoins (use of Bitcoin as a payment system)
<p>CV:</p> <ul style="list-style-type: none"> • Vulnerability to the majority attacks • Loss of private keys accidentally or in case of the death of the owner • Denial of service attacks • Blockchain forks • Transactions latency • Throuput limitation • Energy consumption 	<p>CR:</p> <ul style="list-style-type: none"> • Bitcoin P2P NTW • Wallet applications • Mining process • Internet • Profitability • Strength of Digital signature algorithm • Strength of keys generation

III.4.3 Bitcoin disruption strategies

Malicious attackers who seek to disrupt Bitcoin can either opt for a direct approach or an indirect one:

- A direct approach would target directly Bitcoin COG by influencing the users ‘judgment through lies and propaganda
- An indirect approach would target the Bitcoin CR, which would deny to Bitcoin its CC and therefore lose its brand image and thus lose the trust of its users. This could be achieved through crafting bugs and issues within these CRs. Malicious actions such as owning more than 50% of the hashing power to suppress the consensus mechanism would have a serious impact on transactions confirmation, blocks creation and new currency issuance.

Both a direct and indirect approaches would accomplish the same malevolent purpose, which is the disruption of the system. A best way to secure Bitcoin is to prevent malicious users from exploiting Bitcoin CVs through continuous monitoring and proactive fixes that would strengthen the CRs and protect the COG.

This paper provided an overview of some embedded security features within the Bitcoin ecosystem such as randomness in producing private keys, the elliptic curve discreet logarithm, unbreakable until today, which help produce public keys; the ECDSA that help ensure that TXs are redeemable only by the holders of the private keys; the unbroken properties for the hashing function SHA-256, which are one-way and collision-resistance; the denial of service resistance of the Bitcoin scripting language; the Merkle tree and the back-linkage that help ensure the integrity of the blocks and the Blockchain; the reward of miners which make them work for the system and the difficulty that regulates the mining process according to the network hashing power. The analysis of the different

factors showed that bitcoin center of gravity is the trust that has gained among its users and the consensus mechanism based on the POW that makes the system working properly.

The paper provided also two main strategies to disrupt bitcoin by influencing the judgement of its users as a direct approach and another way that target its critical requirements mainly the mining process which help maintain the system working properly. The paper suggested continuous monitoring and fixes in a proactive manner to strengthen Bitcoin critical requirements. Since many Bitcoin functions are based on this function, such as POW, Merkle tree, TX id, any collision could disrupt the functioning of the system. Hence, Bitcoin community should always monitor the strength of this function and prepare a way ahead to prevent any disruption in case of a collision in SHA-256. Implementation of SHA-512 in the system could serve as alternative.

IV. Conclusion

This thesis paper provided a general overview about the Bitcoin technology. It pointed out some of Bitcoin limitations, such as the throughput of up to seven transactions per second, the confirmation latency, and wastage of energy for miners; security issues, such as zero-confirmation transactions Security, the Blockchain forks issue, transaction malleability, and 51% attack; and applications, such as decentralized storage, identity management, and smart contracts. In addition, it endeavored to examine the relevant risks pertaining to Bitcoin security as a currency and as a payment system. For this purpose, it identified the major feared events for the primary assets, threat scenarios for Bitcoin supporting assets, and the estimated risks using the French risk assessment method known as “*EBIOS*”. Furthermore, this thesis tried to demonstrate the importance of the difficulty feature in Bitcoin security and how it is adjusted dynamically to avoid the Blockchain hijack. Moreover, the research paper suggested also a way to circumvent the 51% attack using a solution of the dining philosophers’ problem that is still to be expanded for a decentralized solution. Also, the researchers carried out a Bitcoin center of gravity analysis through the analysis of Bitcoin critical factors and highlighted some strategies that could be used by malevolent people to disrupt the Bitcoin ecosystem.

A set of security measures were suggested to the Bitcoin community and the Bitcoin users in order to mitigate the estimated risks. Their implementation involve the commitment of the Bitcoin stakeholders at all levels. The Bitcoin users have to be aware of the basics for the security of their wallets and especially the safety of their private keys, which represent a critical asset that secure their funds. Bitcoin nodes should apply preventive security controls such as frequently updating their anti-malware software and thus increasing the security of the peer-to-peer network. Bitcoin developers should continue to fix rising flaws and bugs in the system while considering security as a priority. Also, Bitcoin stakeholders should invest more in mining to increase the security of the Blockchain and the consensus mechanism, which is the cornerstone of the whole system. This latter would protect the Bitcoin system from the greed of some miners who may collude to gain a malevolent majority that could undermine the security of the Blockchain.

Finally, we believe that Bitcoin will persist as the most secure crypto-currency in the market due to the proof-of-work, the hashing functions, the elliptic curve digital signature, and the implication of its developers and their swift reactions to flaws and issues. However, a frequent risk assessment of the technology would help point out major security issues and provide preventive measures in a proactive manner.

The fascinating applications of the Blockchain technology especially for decentralized storage to solve falsification and fraud problems is to be considered for further research. We suggest researchers to look over possibilities of crafting distributed applications that store university diplomas and real-estate titles using the Blockchain technology.

APPENDIXES

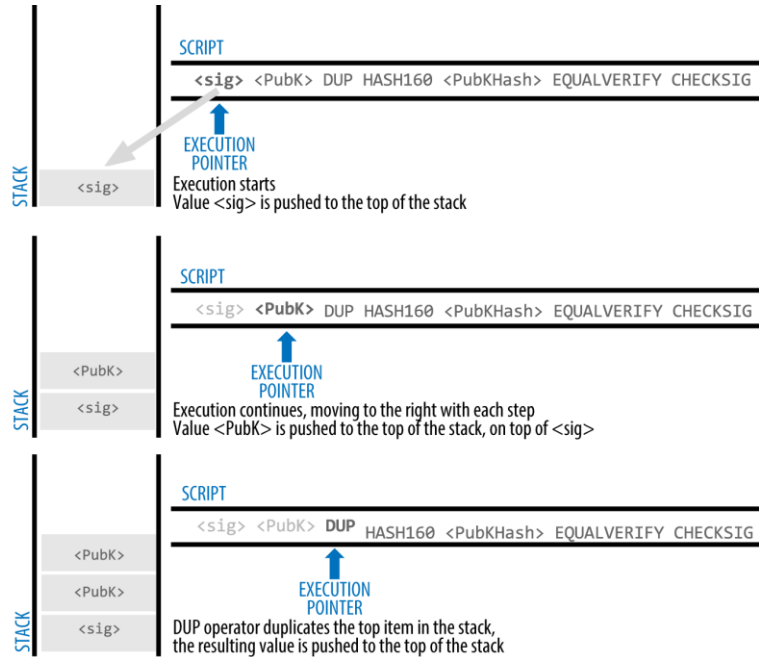
Appendix I

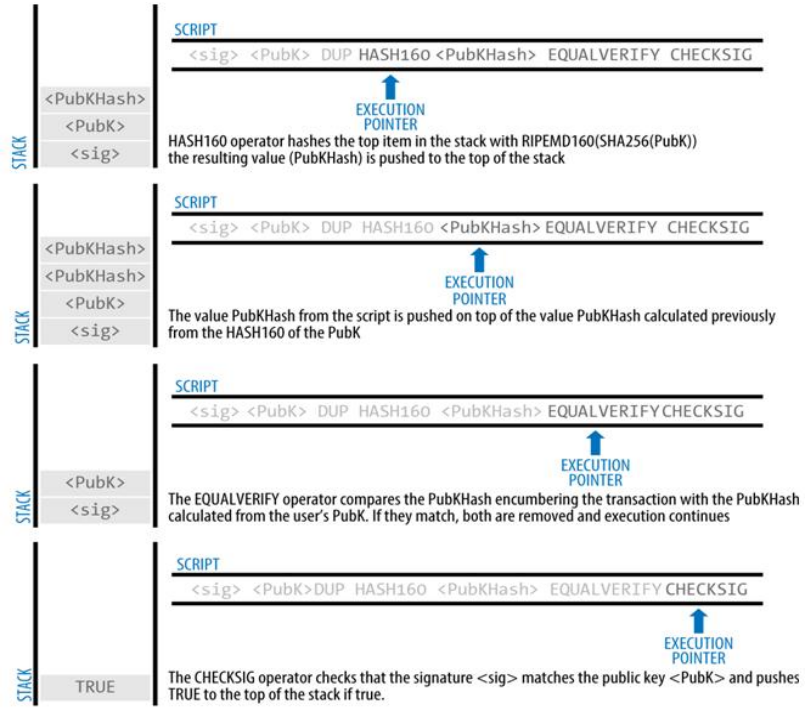
Table I Base58 symbol chart used in Bitcoin [110].

Value	Character	Value	Character	Value	Character	Value	Character
0	1	1	2	2	3	3	4
4	5	5	6	6	7	7	8
8	9	9	A	10	B	11	C
12	D	13	E	14	F	15	G
16	H	17	J	18	K	19	L
20	M	21	N	22	P	23	Q
24	R	25	S	26	T	27	U
28	V	29	W	30	X	31	Y
32	Z	33	a	34	b	35	c
36	d	37	e	38	f	39	g
40	h	41	i	42	j	43	k
44	m	45	n	46	o	47	p
48	q	49	r	50	s	51	t
52	u	53	v	54	w	55	x
56	y	57	z				

Appendix II

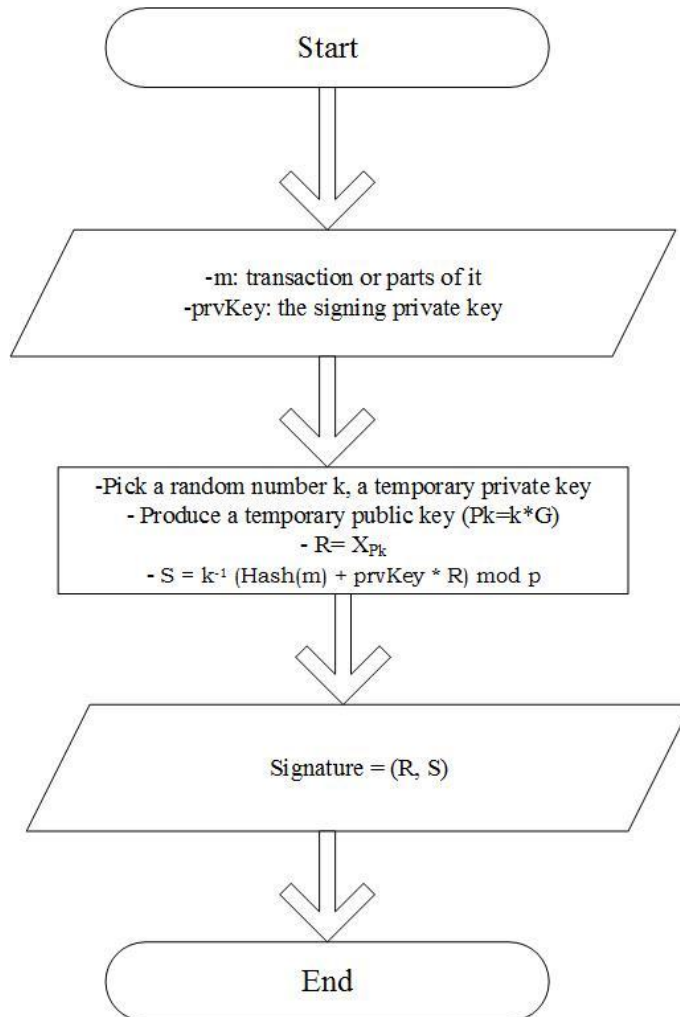
Fig.27 Execution of validation script





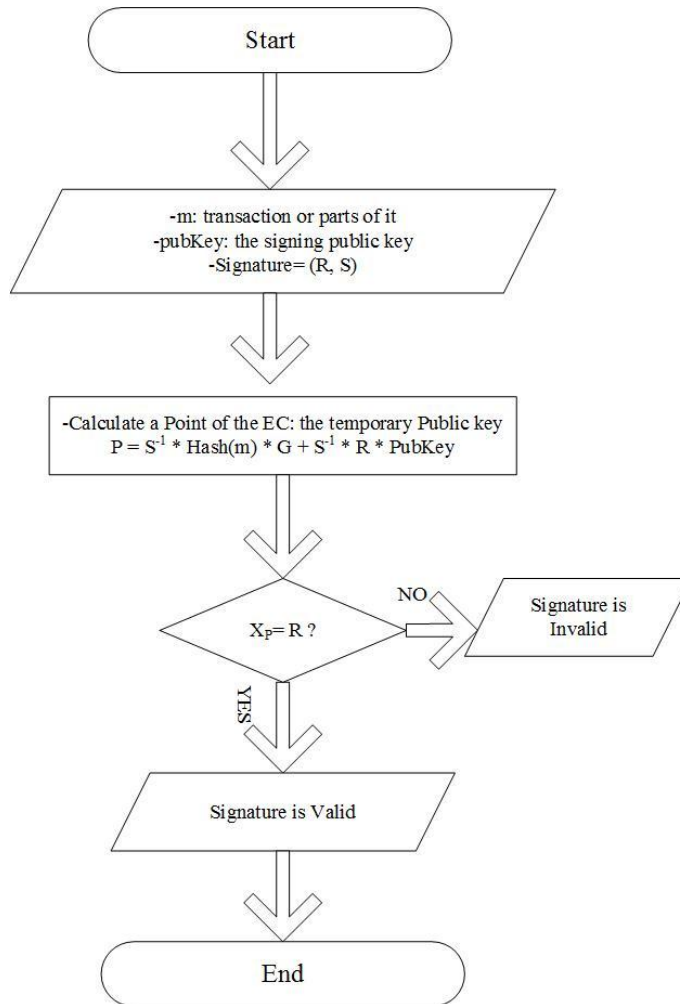
Appendix III

Fig.28 ECDSA signing function used in Bitcoin.



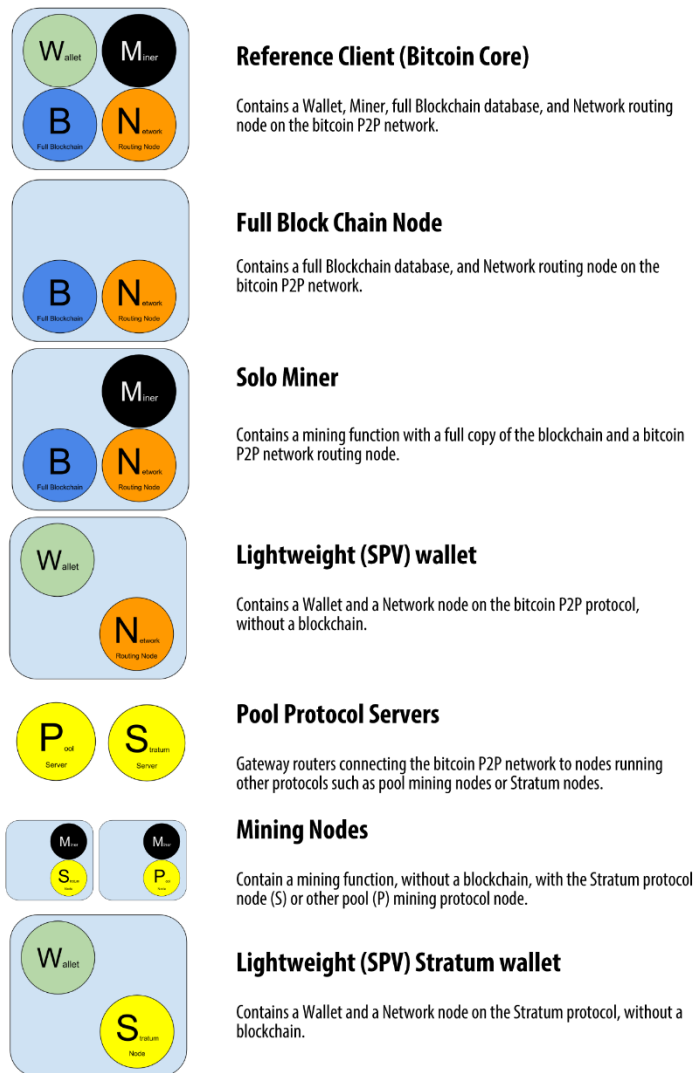
Appendix IV

Fig.29 Verifying function of the ECDSA.



Appendix V

Fig.31 Various types of nodes in an extended Bitcoin network.



Reference Client (Bitcoin Core)

Contains a Wallet, Miner, full Blockchain database, and Network routing node on the bitcoin P2P network.

Full Block Chain Node

Contains a full Blockchain database, and Network routing node on the bitcoin P2P network.

Solo Miner

Contains a mining function with a full copy of the blockchain and a bitcoin P2P network routing node.

Lightweight (SPV) wallet

Contains a Wallet and a Network node on the bitcoin P2P protocol, without a blockchain.

Pool Protocol Servers

Gateway routers connecting the bitcoin P2P network to nodes running other protocols such as pool mining nodes or Stratum nodes.

Mining Nodes

Contain a mining function, without a blockchain, with the Stratum protocol node (S) or other pool (P) mining protocol node.

Lightweight (SPV) Stratum wallet

Contains a Wallet and a Network node on the Stratum protocol, without a blockchain.

Appendix VI

Risk analysis of Bitcoin security using EBIOS method

A. Lamiri¹, K. Gueraoui², G. Zeggwagh³

Abstract – Bitcoin is a virtual currency and a payment system based on cutting-edge innovations in cryptography and distributed systems. It has gained a wide range of popularity since its inception in 2009 and became the most successful crypto-currency in the financial market. It was devised by Satoshi NAKAMOTO and has been developed by many other experts before reaching maturity. Bitcoin security has been looked at by different experts in the past with different levels of analysis and it is still a hot research topic since new improvements are continuously being adopted by the community. Our study provides a high-overview of risks related to Bitcoin as a currency and as a payment system using EBIOS method. **Copyright** © 2010 Praise Worthy Prize Seral. - All rights reserved.

Keywords: Bitcoin, Crypto-currency, EBIOS, Risk Analysis, Security

I. Introduction

Bitcoin is a crypto-currency and an online payment system that does not rely on any central authority or a bank to process the transactions. It is also a protocol that can be used for many other applications beyond the payment system. It was invented in 2008 by an individual or a group of people alias Satoshi NAKAMOTO. They suggested in their white paper, Bitcoin: A Peer-to-Peer Electronic Cash System, a genuine solution to address the double-spending problem using a peer-to-peer network and a hash-based proof of work [1].

Bitcoin took advantage of the cutting-edge advances in cryptography and distributed systems, such as the elliptic curve digital signature, the proof-of-work, and the hashing functions. It was designed as a decentralized and distributed system that works over a peer-to-peer network. This network is made of nodes of participants.

Bitcoin nodes play four different roles, such as network routing, Blockchain database, wallet services, and the mining. They can be full-nodes or lightweight nodes. Full nodes contain a full copy of the Blockchain and can independently verify transactions while lightweight nodes hold only a subset of the Blockchain, mainly the headers of all the blocks that form the Blockchain. Fig.1 illustrates the Bitcoin peer-to-peer network and some of the roles played by the nodes.

The Blockchain is a public ledger made of a chain of blocks in which a block references its previous one, and so on, all the way back to the first block, known as the genesis block. A block is a data structure that contains a

header and a collection of transactions. Each block is identified by its position in the Blockchain, aka the height. It holds also a unique identifier in the form of a digital footprint, known as the blockhash or the proof-of-work (see Fig.2).

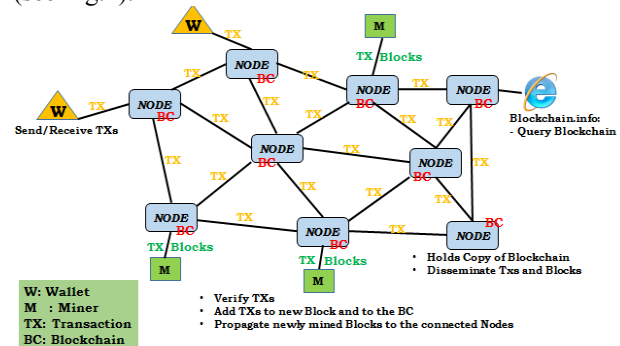


Fig.1. Bitcoin peer-to-peer network and its nodes.

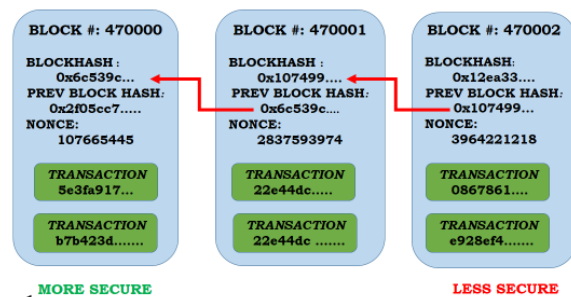


Fig.2. Chain of blocks forming the blockchain.

Blocks are created by miners who compete against each other to find the proof-of-work of the new block that satisfies the target difficulty set by the Bitcoin network.

The target difficulty is set dynamically by the network, every two weeks or 2160 blocks mined, and helps regulate the system so one block can be added every 10 minutes in average. Once blocks are created they are disseminated to the other connected nodes through the Bitcoin peer-to-peer network. The nodes validate them and update their Blockchain. The proof-of-work is the consensus mechanism adopted by the Bitcoin community.

New bitcoins are issued only as a reward for the miner who first calculated the proof-of-work of a new block. This reward is halved every 4 years or every 210,000 mined blocks. By 2140, no bitcoin will be issued, and the amount of money will reach 21 million of bitcoins.

Bitcoin is currently the most successful crypto-currency despite some disruptive fluctuations; its market value is estimated to more than \$10,000 US dollar for one bitcoin [2] and its market capitalization is currently valued to more than \$171 billion US dollar [2]. The Bitcoin flourishing success rises security challenges that the community should continuously monitor and address to preserve its competitive advantage.

Bitcoin security is still a hot research topic since its inception in 2009. Mariam Kiran and Mike Stannett studied different risk areas related to Bitcoin such as social, legal, economic, technological, and security risks. In security risks, they pointed out three high-level risks pertaining to Bitcoin security, which are man-in-the-middle attacks, loss of keys, and the subversive miner strategies [3]. Also, Jerry Brito and Peter Van Valkenburgh mentioned six global threats that could harm the functioning of the Bitcoin. These threats are: flawed key generation, transaction malleability, 51% attacks, Sybil attacks, DDOS attacks, and consensus or fork risks [4]. This paper aims to provide a general overview about security risks related to the Bitcoin by applying a French risk assessment method, known as EBIOS. It is organized as follow: section II lays out the risk assessment methods used for information security and describes the EBIOS method, section III provides details about the risk study and suggests some mitigation techniques, and section IV provides a conclusion for the paper.

II. Risk Assessment Methods for Information Security

II.1 Information Security

Information security is defined as the preservation of confidentiality, integrity and availability of information; in addition, other properties such as authenticity, accountability, non-repudiation and reliability can be involved [5]. This information could be represented in different forms such as paper or electronic devices. Managing information security is a big challenge with the ever-growing threats in the cyber-security realm.

Sustainable organizations assess continuously the risks related to their businesses and the information they rely on to keep their competitive advantage using a risk management process. This process involves four activities, such as risk assessment, risk acceptance, risk treatment, and risk communication.

Risk assessment is the cornerstone step in the risk management process. It involves two techniques, which are: risk analysis and a risk evaluation. While risk analysis aims to identify possible sources of risk, threats or events with harmful impact along with their probability of occurrence, risk evaluation helps determine the significance of the identified risks by comparing them with a set of risk criteria.

II.2 Risk Assessment Methods

There are several information security risk assessment methods available for use. Although they come with different cost and complexity, they tend to achieve the same purpose, which is the analysis and the evaluation of risks pertaining to information security. These methods are but not limited to: MEHARI (CLUSIF, 1997), OCTAVE (CERT, 1999), CRAMM, IRAM, EBIOS (ANSSI, 1995), MAGERIT, etc. There are also some guidelines that address the same issue, such as ISO 27005, NIST SP800-30, Security Risk Management Guide, Australian IT Security handbook, etc.

With all this diversity of methods and guidelines, choosing a method to conduct information security risk assessment may seem challenging. Risk assessment is a resource-consuming task in terms of time, expertise and people involved. A pertinent choice of the appropriate method would save time and frustration. For this purpose, we relied on a comparative study done by Filipe Macedo and Miguel Mira Da Silva. Their study ranked methods such as OCTAVE, EBIOS, MAGERIT, IRAM, IT-Grundschutz, and MEHARI as the most relevant methods in terms of moderate complexity, structured approach, and available tools [6].

All the previously mentioned methods, though differ in complexity and scope, could be used to carry out a comprehensive risk assessment of Bitcoin security and could lead almost to the same conclusions. The main reason we are using EBIOS is because of its adaptability, the availability of its software for free, and its large community of users and supporters. The overarching goal of this paper is to identify and prioritize risks according to their importance and relevance to address the very urgent issues. The following section describes EBIOS risk assessment method and its steps.

II.3 EBIOS

EBIOS stands for “Expression of Needs and Identification of Security Objectives”. It is a free and comprehensive risk assessment method, invented in 1995 by the French National Agency for Information Systems

Security (ANSSI). It is currently supported by a non-profit association of risk management experts, known as EBIOS Club. EBIOS is used by many private and public organizations, in France and abroad, to conduct information systems security (ISS) risk analysis. It helps also produce different security documents such as the security master plan, security policy, protection profile, risk mapping, etc. It is adaptable to different security contexts and can be applied to either basic or complex systems. Furthermore, EBIOS is compliant with major IT security standards [7], such as:

- [ISO/IEC 27001](#): a standard that provides requirements for an information security management system (ISMS).
- ISO/IEC 15408/15443: evaluation criteria for IT security, known also as common criteria.
- [ISO/IEC 17799](#): code of best practices for information security management
- [ISO/IEC 13335](#): management of information and communications technology security. It has currently a status of withdrawn in ISO website [8].
- ISO/IEC 21827: systems Security Engineering Capability Maturity Model.

EBIOS defines risk as a scenario in which risk sources exploit vulnerabilities on the supporting assets which cause incidents for the primary assets. The level of risk is estimated in terms of severity (gravity) and likelihood. Severity is defined as the magnitude of a risk, depends on the level of identification of personal data and the level of consequences or the potential impacts. The likelihood is the feasibility of a risk to occur and it depends on the level of vulnerabilities in the supporting assets.

EBIOS involves a five-stage of an iterative approach as illustrated in Fig.3. The first phase, known as the context study, aims to identify the target system and its environment. It helps specify the issues at stake for the studied system along with the means it uses and the services it must provide. At this stage, all the required information for risk management is collected. This step involves three main activities, which are: definition of study scope, the preparation of the metrics, and the identification of the assets.

Phase 2 is called the feared events (F.E) study. It contributes to the appreciation of risks by identifying and estimating the security needs for the primary assets in terms of confidentiality, integrity and availability. It consists also of identifying the impacts the threat sources if these security requirements were not fulfilled.

Phase 3 involves studying threat scenarios (T.S) that could cause the feared events by determining the threats affecting the supporting assets of the system. It identifies the attacks methods, the threat agents, the vulnerabilities, and the threats levels.

Phase 4 aims to estimate and assess the risks affecting the system and identify options to treat them. Finally,

phase 5 aims to determine the security measures to be put into action and analyze the residual risks.

In the next section, we follow these five steps of EBIOS method to determine security risks pertaining to Bitcoin as a currency and as a payment system.

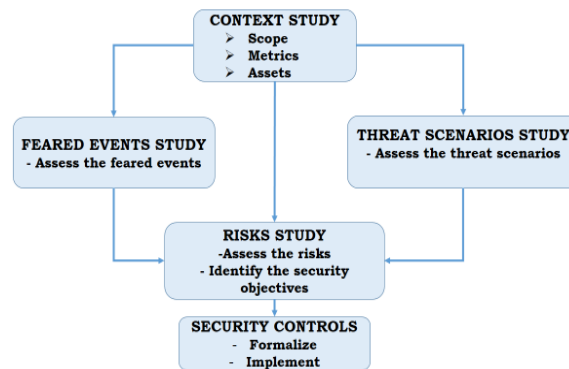


Fig.3. Phases of EBIOS Method.

III. Risk Study

In this section, we will apply the EBIOS method for Bitcoin to determine its major security risks. First, we will determine the context of the study; then identify the feared events and the threats scenarios followed by an analysis of the identified risks and finally suggest some security measures to address these risks.

III.1. Context Of The Study

This study aims to assess information security risks pertaining to Bitcoin. The goal is to manage, in a proactive approach, the risks that could eventually undermine Bitcoin. It will identify threat scenarios for the Bitcoin supporting assets, feared events for the primary assets, and threats and vulnerabilities. This study should highlight security measures designed to minimize the identified risks.

The focus of this study is the information security risks related to Bitcoin as a currency and as a payment system. The main participants in the Bitcoin security are:

- Users and the stakeholders;
- Miners;
- Nodes in the Bitcoin peer-to-peer network
- Bitcoin’s community of developers.

The Bitcoin’s challenges are:

- Stay available for use around the clock 24/7;
- Include and validate all the valid transactions in the Blockchain every 10 minutes in average;
- Keep a clean and safe copy of the Blockchain in most nodes;
- Solve continuously the fork-issue to avoid the double-spending problem.

This study concerns only the information security risks related to Bitcoin. It excludes the following risk areas:

- Social risks.
- Legal risks.
- Economic risks.

The study will examine the risks pertaining to the use of Bitcoin, mainly risks related to sending or receiving transactions, creating a new block and calculating its proof-of-work, propagating transactions and blocks to the connected nodes, storing a full and a clean copy of the Blockchain in most of the network’s nodes, and improving the Bitcoin protocol and upgrading it so to continue addressing future needs. Fig.4 depicts in detail the perimeter of the study.

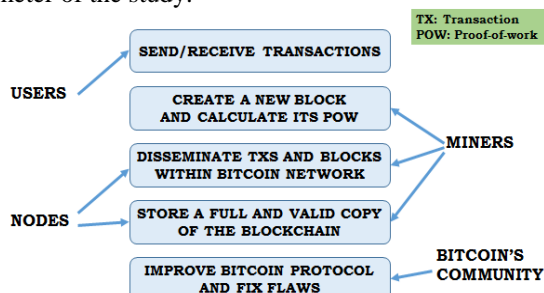


Fig.4. perimeter of the study.

The plausible threat sources in the context of our study are shown in Table I as follows:

TABLE I
THREAT SOURCES

Threat source type	Threat sources
External human source, malevolent, with unlimited capabilities	State-sponsored attacks in the form of an APT
Internal human source, not malevolent, with weak capabilities	Imprudent user
External human source, malevolent, with significant capabilities	Hacker of group of highly skilled hackers competitors
Internal human source, not malevolent, with significant capabilities	Less serious administrator Less serious employee
External human source, malevolent, with weak capabilities	Cleaning personnel/Janitor Thief
Malevolent software of unknown origin	Flaw in the application Non-targeted virus
Natural phenomenon	Breakdown of material or network
Natural or health disaster	Earthquake/Fire/Flood/Tornado Illness or accident
Internal human source, malevolent, with unlimited capabilities	Greedy miner Malevolent administrator Subversive miners
Internal human source, malevolent, with significant capabilities	Malevolent Bitcoin developers Malevolent employees Corrupted nodes

The preparation of the metrics aims to define a collection of parameters and scales that will serve to manage the risks related to Bitcoin, such as the security criteria and the scales of security needs. These criteria are factors that gauge the importance of different primary assets according to the business needs. The three unavoidable security criteria are defined as follow:

• **Confidentiality (C):** a property meaning that primary assets are accessible only to authorized personnel. In this context, the objective is to protect the identity of the Bitcoin user

• **Integrity (I):** a property of exactness and completeness of the primary assets. This means that primary assets are not altered;

• **Availability (A):** a property meaning that the primary assets are accessible at any giving time;

In this study we are using the following scale levels (see Table II) for the retained security criteria.

TABLE II
THE RETAINED SECURITY CRITERIA AND THEIR SCALE LEVELS

Security criteria	Scale level	Detailed description
Confidentiality	1. Public	The primary asset is public.
	2. Limited	The primary asset must only be accessible to staff and partners.
	3. Reserved	The primary asset must only be accessible to the (internal) staff involved
	4. Private	The primary asset must only be accessible to people who have been identified and who need to know
Integrity	1. Detectable	The primary asset can be corrupted but the alteration can be identified.
	2. Controlled	The primary asset can be corrupted, if the alteration is identified and the integrity of the primary asset can be restored
	3. Has integrity	The primary asset must be rigorously uncorrupted.
Availability	1. More than 48h	The primary asset can be unavailable for more than 48 hours.
	2. Between 24h and 48h	The primary asset must be available within 48 hours
	3. Between 4h and 24h	The primary asset must be available within 24 hours
	4. Less than 4h	The primary asset must be available within 4 hours

For risk assessment, we need also to establish two scales: a scale of severity (gravity) and a scale of

likelihood. A scale of severity describes all the possible levels of impact. Table III shows the scale levels retained for evaluation of severity.

TABLE III
SCALE LEVELS OF SEVERITY

Scale level	Detailed description
1. Negligible	Bitcoin will overcome the impact with no difficulty
2. Limited	Bitcoin will overcome the impact despite some difficulty
3. Important	Bitcoin will overcome the impact with serious difficulty
4. Critical	Bitcoin will not overcome the impact (its survival is threatened)

The scale of likelihood, as illustrated in Table IV, describes all the possible levels of likelihood of the threat scenarios.

TABLE IV
THE LIKELIHOOD SCALE LEVELS.

Scale level	Detailed description
1. Minimal	This have not to recur
2. Significant	This could recur
3. Strong	This should recur
4. Maximal	This will certainly recur in the future

At the beginning of any risk analysis study, EBIOS requires the establishment of risk management criteria, which are rules that help make decisions throughout the study. These criteria help estimate and evaluate the risks. Table V illustrates some of the retained criteria

TABLE V
RISK MANAGEMENT CRITERIA.

Action	Risk management criteria
2.1.1. Analysis of all the F. E	The F.E are estimated in terms of severity according to the defined scale of levels.
2.1.2. Assess each F. E	The F.E are ranked in a decreasing order of likelihood.
3.1.1. Analysis of all the T.S.	T.S are estimated in terms of their likelihood according to the defined scale of levels
3.1.2. Assess each T. S	The T.S are ranked in a decreasing order of their likelihood.
4.1.1. Analyze the risks	The severity of a risk is equal to the considered F.E. The likelihood of a risk is equal to the maximal likelihood of all the T.S linked to the considered F.E.
4.1.2. Assess the risks	Risks of critical severity and those of important severity with strong or maximal likelihood are to be considered intolerable. Risks of important severity and significant likelihood, and those of limited severity and strong or maximal likelihood are to be considered as very significant. Risks of important severity and minimal likelihood or those of limited severity with a significant likelihood are to be considered significant.
4.2.1. Choose options for risk treatment	Intolerable, very significant, and significant risks must be reduced to an acceptable level, transferred or avoided if this is possible. Negligible risks can be accepted.
5.1.1. Determine the controls	Security measures must be selected according to the context to minimize or eliminate the threat scenarios by fixing a vulnerability or by limiting the impact.

At this stage, we need to identify the assets, within the perimeter of the study, mainly the primary and the supporting assets. The primary assets (P.A) represent the informational assets or the immaterial assets that we want to protect. In other terms, this means the assets for which

Primary assets	Description
Private keys	These keys are used in the digital signature required to spend transactions
Transaction (TX)	Transactions consist of two main parts, which are the input and the output. The input of a transaction contains an unlocking script, which is mostly a digital signature and a public key.
Block	The block is a data structure made of a header and a collection of transactions.
Blockchain	The Blockchain is a distributed database that stores all the transactions in the Bitcoin system.
Consensus mechanism	The consensus is achieved through the calculation of the proof of work of the new block, which is disseminated to the connected nodes for validation.

the non-respect of security criteria (C.I.A) will put them in danger. Table VI lists the major primary assets.

TABLE VI
THE STUDY'S MAJOR PRIMARY ASSETS.

The supporting assets (S.A) are the technical or non-technical components of the studied system, which support the primary assets. The study of these assets is important since they may hold some vulnerabilities that the threat sources might exploit to hurt the security of the primary assets. Table VII illustrates the supporting assets.

TABLE VII
THE STUDY'S MAJOR SUPPORTING ASSET

Supporting asset	Type
Mining sites	Premises
Wallet applications	Desktop wallet
	Mobile wallet
	Online wallet services
Internet	Networks
Bitcoin peer-to-peer network	
Hardware wallet	Hardware
User's device	
Bitcoin nodes' machines	
Mining machines	
Bitcoin user	People
Paper wallet	Paper

EBIOS requires to establish the relationship between the primary and supporting assets so risks within the perimeter of the study can be compiled later in phase 4.

After selecting the assets that will be considered in our study, we should carry out a census of the existing security measures for the supporting assets. These are technical or non-technical controls that can be categorized in three types, which are:

- Preventive controls are measures that protect vulnerabilities and make an attack ineffective or decrease its impact;

- Protective controls are measures that discover attacks and activate preventive or corrective controls;
- Recovery (Restoration) controls are measures that are often associated with business continuity and disaster recovery.

Table VIII illustrates some of the existing security measures for Bitcoin. It may not be complete, but it provides the basics of security controls that are or should be implemented in the studied system.

TABLE VIII
EXISTING SECURITY MEASURES

Label	Associated supporting asset	Category of the measure		
		Preventive	Protective	Recovery
Security of the premises	Mining sites	X	X	
Air-conditioning		X		
Fire-fighting devices		X		
Access control using password	Wallet applications	X	X	
Wallet backups				X
Anti-malware solutions	User's device	X	X	
Network service security	Bitcoin peer-to-peer network	X	X	
Business continuity plan	Mining machines			X
Anti-malware solutions		X	X	
Anti-malware solutions	Bitcoin nodes' machines	X	X	

III.2 Study Of The Feared Events

At this stage of EBIOS, we identify the generic scenarios that we wish to avoid within the perimeter of the study. The thought process is done at the functional level rather than the technical level, which means that the focus is on the feared events affecting the primary assets and not on those impacting the supporting assets.

Then we identify the feared events that are affecting the primary assets, for each security criterion. After that, we list the security needs for each primary asset, the impact in case of non-respect of security measures and the related threat sources along with a level of severity. Table IX illustrates the feared events, selected in our study, which are related to the Confidentiality, the Integrity, and the Availability of the transaction, the block, the Blockchain, the consensus mechanism, and the private keys.

Assessing the feared events involves judging how important these events are within the perimeter of the study taking into consideration the established risk management criteria. For this purpose, the feared events are then prioritized according to their severity. This study identified twelve feared events as illustrated in Table X.

P - A	Security criteria	Security requirements	T. S	Impacts	Severity
T X	C	Limited	Imprudent user	Reputational damage	Limited
			Malevolent bitcoin developers	Putting someone in danger	
			Flaw in the application	Loss of credibility with users	
				Loss of anonymity	

TABLE IX
EXCERPT OF THE FEARED EVENT LIST

TABLE X
THE IDENTIFIED FEARED EVENTS

Severity (gravity)	Feared events
Critical	<ul style="list-style-type: none"> ▪ Block – availability ▪ Block – integrity ▪ Consensus mechanism - availability ▪ Private keys - availability ▪ Private keys - confidentiality ▪ Private keys - integrity ▪ Transaction - availability
Important	<ul style="list-style-type: none"> ▪ Blockchain – availability ▪ Blockchain – integrity ▪ Consensus mechanism - integrity ▪ Transaction - integrity
Limited	<ul style="list-style-type: none"> ▪ Transaction - confidentiality
Negligible	
Not retained	<ul style="list-style-type: none"> ▪ Block – confidentiality ▪ Blockchain – confidentiality ▪ Consensus mechanism - confidentiality

III.3. Study Of Threat Scenarios

This step of the EBIOS method involves identifying the generic threat scenarios that could harm the information security of Bitcoin within the established perimeter of the study. These threat scenarios affect chiefly the supporting assets and not the primary assets. For this purpose, the thought process is carried out, at the technical level rather than the functional level.

First, we identify the threat scenarios affecting each supporting asset and for each security criterion and then estimate their likelihood. We consider all the elements that participate in the threat scenarios such as the threats, the vulnerabilities, and the threat sources. An excerpt of the result of this study is shown in Table XI

TABLE XI
EXCERPT OF THE THREAT SCENARIOS LIST

S. A	Security criterion	T. A	Threat
Paper Wallet	Confidentiality	<ul style="list-style-type: none"> ▪ Imprudent user ▪ Thief 	<ul style="list-style-type: none"> ▪ Spying a paper wallet ▪ Wear of a paper wallet ▪ Loss or theft of a wallet paper
	Vulnerabilities	Prerequisite	Likelihood
	<ul style="list-style-type: none"> - Allows the observing of interpretable data - Poor quality constituents - Not suitable for the conditions of use - Portable 	<ul style="list-style-type: none"> - Knowledge of the existence and location of the paper media - Physical access to the paper media (legitimate or illegitimate accessing, or bypassing) 	Maximal

Assessing a threat scenario means judging its importance within the perimeter of the study while taking into consideration the established risk management criteria. The identified threat scenarios are ranked as shown in Table XII.

TABLE XII
EVALUATION OF THREAT SCENARIOS

Likelihood Level	Threat Scenarios
Maximal	Bitcoin users – availability Paper wallet - confidentiality
Strong	Bitcoin nodes – availability Mining machines – availability Paper wallet – availability Users device – availability Users device – confidentiality Wallet applications - confidentiality
Significant	Bitcoin nodes – integrity Bitcoin peer-to-peer network – availability Bitcoin peer-to-peer network – integrity Bitcoin users – confidentiality Hardware wallet – availability Hardware wallet – confidentiality Hardware wallet – integrity Mining machines – integrity Wallet applications – availability Wallet applications – integrity
Minimal	Bitcoin users – integrity Paper wallet – integrity Users device – integrity
Not retained	Bitcoin nodes – confidentiality Bitcoin peer-to-peer network – confidentiality Mining machines – confidentiality

III.4 Study Of Risks

At this phase, we assess the risks related to Bitcoin and then identify the security objectives, which determine the way to address these risks. Analyzing the risks involves identifying those risks affecting the perimeter of the study

along with their severity and their likelihood in two steps. In the first step, we do not take into consideration the existing controls while in the second step we take them into account.

Risk analysis implies linking the feared events (FE) and the threat scenarios (TS). EBIOS suggests two ways to establish this linkage, which are:

$$R (\text{Risk}) = 1FE + 1TS \quad (1)$$

$$R (\text{Risk}) = 1FE + TS1 + TS2 + \dots + TS_n \quad (2)$$

The first formula suggests a risk for each feared event and each threat scenario. This formula leads to multiple risks, which can be cumbersome. The second formula calculates a risk for each feared event and a set of threat scenarios, which may be more pertinent since feared events directly impact the primary asset that we want to protect. For these reasons, we chose the second formula. After applying it, we ended up with 12 risks, as numerous as the feared events (see Table XIII). Table XIV illustrates these risks along with their estimation without and with security measures (SM)

TABLE XIII
RELEVANT RISKS TO BITCOIN SECURITY

Risk Label	Threat Scenarios
R0–risk related to TX (A)	Wallet app (A)/Users device (A) p2p network (A)/nodes (A) Mining machines (A)/users(A)
R1 – risk related to TX (I)	Wallet app (I)/Users device (I) p2p network (I)/nodes(I) Mining machines (I)/users (I)
R2–risk related to TX (C)	Wallet app(C)/User device (C) users (C)
R3–risk related to block (A)	p2p network(A)/ nodes(A) Mining machines (A)
R4 – risk related to block (I)	p2p network(I)/nodes(I) Mining machines (I)
R5–risk related to Blockchain (A)	Wallet app (A)/Users device (A) users (A)/p2p network (A) nodes (A)/Mining machines (A)
R6–risk related to Blockchain (I)	Wallet app (I)/Users device (I) p2p network (I)/ nodes (I) Mining machines (I)/ users (I)
R7–risk related to consensus mechanism (A)	Wallet app (A) Users device (A)/p2p network (A) nodes (A)/Mining machines (A)
R8–risk related to consensus mechanism (I)	Wallet app (I) Users device (I)/p2p network(I) Bitcoin nodes (I)/Mining machines (I)
R9–risk related to private keys (A)	Wallet app (A)/ Users device (A) Users (A)/Hardware wallet (A)/Paper wallet (A)
R10–risk related to private keys (I)	Wallet app (I)/Users device (I)/ users (I) Hardware wallet (I)/Paper wallet (I)
R11–risk related to private keys (C)	Wallet app (C)/User device (C) users (C)–Hardware wallet (C) Paper wallet(C)

TABLE XIV
RISKS ESTIMATION

Risk	Estimation without SM		Estimation with SM	
	Severity	Likelihood	Severity	Likelihood
R0	Critical	Maximal	Limited	Significant
R1	Important	Significant	Limited	Significant
R2	Limited	Strong	Limited	Minimal
R3	Critical	Strong	Limited	Significant
R4	Critical	Significant	Limited	Minimal
R5	Important	Maximal	Important	Minimal
R6	Important	Significant	Important	Minimal
R7	Critical	Strong	Important	Significant
R8	Important	Significant	Limited	Minimal
R9	Critical	Maximal	Important	Minimal
R10	Critical	Significant	Limited	Minimal
R11	Critical	Maximal	Important	Minimal

Risk Assessment involves judging the importance of the risks according to the pre-established risk management criteria. Some of these risks can be omitted if they are deemed weak. Fig. 5 illustrates the evaluation of risks after taking into consideration the security measures.

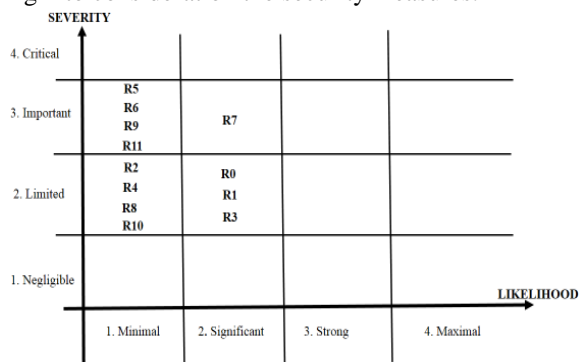


Fig.5. Risk assessment illustration

We assess the identified risks according to the risk management criteria established at the beginning of this study. For this purpose, we consider:

- **Intolerable Risks:** are those of critical severity and those of important severity with strong or maximal likelihood. In our case, we do not have any intolerable risks.
- **Very Significant Risks:** are those of important severity and significant likelihood, and those of limited severity and strong or maximal likelihood. In our case, R7 is a very significant risk.
- **Significant Risks:** are those of important severity and minimal likelihood or those of limited severity with a significant likelihood. In our case, these risks are: R0, R1, R3 and R5, R6, R9, R11.
- **Negligible Risks:** are those of limited severity and minimal likelihood. These are R2, R4, R8, and R10.

The analysis of risks pertaining to Bitcoin security showed four types of risks that should be addressed according to the established risk management criteria. At this stage, we should identify the options to treat these

risks. There are four options to treat a risk, which are: Avoid (or refuse), Reduce, Accept, Transfer (or share).

According to the retained risk management criteria, intolerable, very significant, and significant risks must be reduced to an acceptable level, transferred or avoided if this is possible. Negligible risks can be accepted. Therefore, risks such as R0, R1, R3, R5, R6, R7 and R9 must be reduced to an acceptable level while risks such as R2, R4, R8, and R10 can be accepted. R11 can be either reduced or transferred to a third party such as an insurance company to share the risk of loss or theft. Table XV illustrates the security objectives for this study.

TABLE XV
SECURITY OBJECTIVES

Risk	Severity	Likelihood	Security Objective
R0	Limited	Significant	Reduce
R1	Limited	Significant	Reduce
R2	Limited	Minimal	Accept
R3	Limited	Significant	Reduce
R4	Limited	Minimal	Accept
R5	Important	Minimal	Reduce
R6	Important	Minimal	Reduce
R7	Important	Significant	Reduce
R8	Limited	Minimal	Accept
R9	Important	Minimal	Reduce
R10	Limited	Minimal	Accept
R11	Important	Minimal	Reduce or transfer

After achieving every security objective, some of the risks might remain and should also be addressed. These are called residual risks that we must address with complementary security measures. For this purpose, we are going to consider the following rules:

- Avoided risks do not generate residual risks if they were completely avoided.
- Reduced risks lead to residual risks if they are not completely reduced.
- Accepted risks are entirely residual risks.
- Transferred risks do not imply residual risks.

These residual risks that will remain after the implementation of the security measures have also to be estimated in terms of severity and likelihood. Table XVI illustrates the major residual risks along with their evaluation.

TABLE XVI
MAJOR RESIDUAL RISKS AND THEIR EVALUATION

Residual risk	Severity	Likelihood
Risk linked to the compromise of the Blockchain	Important	Minimal
Risk related to the disclosure of the Private Keys	Important	Minimal

In the next section, we determine the security measures that would lead to the achievement of the security objectives.

IV.5. Security Controls

At this stage of the EBIOS method, we determine the security measures that will allow us to achieve the security objectives, which means we need to highlight those controls that will allow us to avoid, reduce or transfer some of the identified risks.

Table XVII presents a list of security measures destined to address the major risks related to Bitcoin in accordance with the security objectives.

TABLE XVII
SECURITY MEASURES DESTINED TO REDUCE OR TRANSFER THE RISKS RELATED TO BITCOIN

Security Measure	Risks										
	R0	R1	R3	R5	R6	R7	R9	R11			
M1- Wallet Backups Encryption	X	X			X	X	X	X			
M2- Monitoring and re-examination of third party service	X			X		X	X	X			
M3- Use wallets that implement mnemonic sentence (wallets that implement BIP-39)	X			X	X	X					
M4- Protection against exterior and environmental threat	X		X	X	X	X					
M5- Restrict visits in the Mining sites	X		X		X	X					
M6- Sensitive Assets inventory	X		X	X	X	X					
M7- Management of removable media											
M8- Make aware, qualification and training in matters of security	X		X	X	X	X	X	X			
M9- Withdrawal of access right	X			X	X	X	X	X			
M10- Use of anti-spying screens	X			X		X	X	X			
M11- Choice of location and protection of hardware-use of anti-theft systems	X			X	X	X	X	X			
M12- Safe-guarding paper wallet from loss or theft								X	X		
M13- Measures against malevolent code- frequent update of anti-virus	X	X	X	X	X	X	X	X			
M14- Alternative power suppliers on case of power outage	X	X	X	X	X	X					
M15- Sensitization about basics security measures	X	X		X	X		X	X			
M16- Contracting an insurance to address private keys loss or theft								X	X		

The best way to mitigate risks is to adopt a strategy of defense in depth, which relies on three layers of defense. These layers are but not limited to: preventive layers, protective layers, and recovery layers. Table XVIII illustrate these layers of defense.

TABLE XVIII
LAYERS OF DEFENSE RELATED TO THE SECURITY MEASURES

Security Measure	Associated supporting asset	Category of the measure		
		Preventive	Protective	Recovery
M1	Wallet applications	X	X	
M2				
M3	Wallet applications Users device	X		X
M4	The mining sites Bitcoin P2P network Mining machines	X		
M5	The mining sites Mining machines	X		
M6		X		
M7				
M8	The mining sites Wallet applications (Personnel)	X		
M9		X		
M10	User's device	X		
M11	User's device Hardware wallet	X		
M12	Paper wallet	X		
M13	Users device Mining machines Hardware wallet Bitcoin nodes	X	X	
M14	Mining machines	X		
M15	Bitcoin users	X		
M16	Paper wallet Wallet applications Hardware wallets	X		

This list of measures was mostly adapted from the ISO 27002 standard. These measures, if they are correctly implemented, would help decrease the severity and the likelihood of the most identified risks pertaining to Bitcoin. However, two residual risks may remain and should be monitored so they cannot harm the functioning of the whole system. These residual risks are:

- Risk linked to the compromise of the Blockchain
- Risk related to the disclosure of the Private Keys

The underlying assumption of Bitcoin security was based on the honesty of the Miners. These miners gain more in staying honest through the reward they receive

for new mined blocks and through the transactions fees they collect. Although this assumption would stay strong for the future, Bitcoin stakeholders should invest in mining to keep most of the hashing power and therefore ensure more security for their bitcoins. This measure combined with frequent monitoring and risk assessment would reduce the risk of the compromise of the Blockchain to an acceptable level. Also, transferring the risk related to the disclosure of the private keys to an insurance company would decrease its impact and by consequence keep this risk in a tolerable level

V. Conclusion

This study demonstrated that EBIOS is an adaptable method that could be used for different contexts. Indeed, EBIOS was useful to carry out a risk assessment for the most successful crypto-currency, the Bitcoin. In this paper, we examined the relevant risks pertaining to Bitcoin security as a currency and as a payment system. We identified the major feared events for the primary assets and the threat scenarios that are threatening the Bitcoin supporting assets. We then combined the feared events and the threat sources to calculate and estimate the risks. This study identified twelve risks pertaining to Bitcoin security, which are:

- Risks related to transaction confidentiality Integrity and availability;
- Risks related to block integrity and availability;
- Risks related to Blockchain integrity and availability;
- Risks related to consensus mechanism integrity and availability;
- Risks related to private keys confidentiality integrity and availability;

These risks were examined in detail and ranked according to their severity and likelihood without and with pre-existing security measures.

At the end of the study, we suggested some extra security measures that would allow achieving the security objectives, so we could avoid, reduce or transfer some of these identified risks. These security measures were suggested in three different layers of defense consisting of preventive, protective, and recovery controls. Their implementation involves the commitment of the Bitcoin users and stakeholders at all levels.

The study concluded that despite the security measures, two residual risks would continue to threaten the Bitcoin security. These risks are:

- Risk linked to the compromise of the Blockchain;
- Risk related to the disclosure of the Private Keys.

Frequent monitoring of the Blockchain and a transfer of the risk related to the disclosure of the private keys to an insurance company would decrease the impact of these two residual risks.

Furthermore, we suggest the following:

- The Bitcoin users must be aware of the basics for the security of their wallets and especially the safety of their private keys, which represent a critical asset that secure their funds. Bitcoin nodes should apply preventive security controls such as frequently updating their anti-malware software and thus increasing the security of the peer-to-peer network.
- Bitcoin developers should continue to fix rising flaws and bugs in the system while considering security as a priority.
- Bitcoin stakeholders should invest more in mining to increase the security of the Blockchain and the consensus mechanism, which is the cornerstone of the whole system. This measure would protect the Bitcoin system from the greed of some miners who may collude to gain a malevolent majority that could undermine the security of the Blockchain.

Finally, a frequent risk assessment of the technology would help point out major security issues and provide preventive measures in a proactive manner. For this purpose, we suggest the Bitcoin community to adopt EBIOS for risk assessment and to establish a knowledge base tailored EBIOS software for cryptocurrency risk analysis.

Appendix VII

Bitcoin difficulty, a security feature

Abdenaby Lamiri^{1,1}, Kamal Gueraoui¹, Gamal Zeggwagh¹,

¹ Modeling and Simulation in Mechanics and Energetics Team (MSME) of the Research Center on Energy, Mohamed 5th University, Rabat.

{abdamsic@gmail.com, kgueraoui@yahoo.fr, gamalzeggwagh972@hotmail.com}

Abstract. Bitcoin has been growing, since its inception in 2009, to gain a financial mainstream despite the constant fluctuations in its value. It is currently ranked as the most successful Crypto-Currency and decentralized payment system among the others. This success is due, to some extent, to its security, which depends mainly on the cutting-edge cryptographic innovations, such as the hashing functions, the elliptic curve digital signature (ECDSA), and the difficulty that regulates the mining process and allows the system to keep up with the increasing hash-rate. This paper provides an overview of Bitcoin difficulty and how it contributes to the security of this Crypto-Currency.

Keywords: Bitcoin, Blockchain, Crypto-Currency, Difficulty, Security.

1 Introduction

Since 2009, Bitcoin value has been soaring in a rapid rate until it reached a peak on December 17th, 2017[111] when it attained, for the first time in history, more than \$ 20,000 US dollars. Since then, its value has decreased tremendously, and it is currently fluctuating around \$ 7, 000 US dollars for one bitcoin.

Bitcoin is a decentralized system that does not rely on any third party to process the transactions. Transactions are collected and validated by all the participating nodes connected to the peer-to-peer network. The validated transactions are stored in blocks and these blocks are added to the Blockchain. The Blockchain is a distributed ledger that contains all the valid transactions that ever happened in the system.

Bitcoin is considered a secure system since it relies on the implementation of some of the advanced cryptographic features. For instance, the Bitcoin keys and addresses generation process is secure because of the randomness used in producing the private keys, the elliptic curve discrete logarithm, which cannot be solved giving the current computational power, the one-way property of the hashing functions SHA256 and RACE MD (RIPEMD160), and because no backdoors were yet discovered in the elliptic curve. This elliptic curve used in Bitcoin is defined by a standard known as SECP256K1 [112].

This security is improved using the Base58Ckeck formatting, which ensure the integrity of the Data, mainly for the Bitcoin keys and addresses [113]. Also, the use of the ECDSA ensures that only the holders of the private keys can redeem the related funds. Other security features are added to Bitcoin to ensure the most prominent security objectives such as the confidentiality, the integrity and the availability. These features are, for instance, the back-linkage of blocks, which helps ensure the integrity of the Blockchain; the Merkle tree, which ensures the Block integrity; and the difficulty, which ensures the system integrity since it forces miners to work hard for at least 10 minutes to find a proof-of work for a new block.

This paper aims to provide some insights about the Bitcoin difficulty by illustrating how it is related to the target and the Bits value, and how it contributes to the Bitcoin security. It provides some scripts, written in python 3, to calculate the difficulty and the target and to verify the proof-of-work. It shows also the correlation between the difficulty and the hashing rate used in Bitcoin.

2 Related Works

Since its inception in 2009, Bitcoin security became an active research area that interested many researchers around the

world. Juan Garay, Aggelos Kiayias, and Nikos Leonardos studied Bitcoin difficulty and suggested a Bitcoin protocol with chains of variable difficulty as a way to deter any malicious adversary controlling a fraction of miners holding around 50% of the mining power [114]. Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert van Renesse mentioned that the difficulty provides some resilience to mining power variation by allowing different instances of blockchain to tune their proof of work difficulty at different rate in order to maintain a stable rate of Blocks [115].

3 Bitcoin Difficulty

Difficulty can be defined as a measure of how difficult it is to find a hash (proof-of-work) below a given target [116]. This parameter is set dynamically by the Bitcoin network every 2016 blocks or two-weeks in average. The difficulty is tied to two other parameters, the target and the Bits, which we will explain in the following paragraphs.

The proof-of-work serves as a proof that the miner has committed a great amount of hashing power to find the block header's hash that satisfies the required condition. The proof-of-work is hard to find but easy to verify. It involves finding a value for the nonce that results in a block's header hash, using SHA-256 algorithm, that is less or equal to the difficulty target (target). So how this target is calculated?

Every block contains a field called "Bits", known also as target Bits, which is a four-byte number represented in a hexadecimal floating-point format. Bits value serves to calculate the difficulty target, which is used as a condition in the mining algorithm. The Bits field value of the first block in the Blockchain is 1d00ffff [117]. By convention, the first two digits (1d) represent the total number of digits a target is made of. It is used in the exponent of the floating-point notation while the remaining digits (00ffff) represents the coefficient. Now, how the target is derived from the Bits value?

To calculate the target from the Bits value, we rely on the following formula:

$$\text{TARGET} = \text{COEFFICIENT} * 2^{8 * (\text{EXPONENT} - 3)} \quad (1)$$

Where:

- *COEFFICIENT* is the three Bytes on the right part of 4-Byte format of the Bits.
- *EXPONENT* is the first Byte on the left part of 4-Byte format of the Bits.

Using the hexadecimal representation and applying this formula to the block #0 with Bits value of (0x1d00ffff), the target would be:

$$\text{TARGET} = \text{0X00FFFF} * 2^{8 * (\text{0X1D} - \text{0X3})} \quad (2)$$

Therefore, the result in hexadecimal format is:

TARGET (in HEX) =

0xffff00

We compare the header's hash of the Block #0 (proof-of-work of Block #0) with the calculated target, using python 3. The following Script shows that the Block Header's Hash is less or equal the calculated target, which means that the proof-of-work (POW) is valid.

```
>>> #calculating the target of Block #0 using the Bits Value
>>> Target = 0x00ffff*2**(0x8*(0x1d-0x3))
>>> # the decimal number is:
>>> print (Target)
26959535291011309493156476344723991336010898738574164086137773096960
>>> # the Target in hexadecimal representation
>>> hex(Target)
'0xffff000000000000000000000000000000000000000000000000000000000000'
>>> # Now let's compare this target to the block #0 header's hash
>>> BlockHash=0x000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
>>> BlockHash <= Target
True
```

The target condition sets the frequency at which a new proof-of-work is found. It determines also the difficulty for a collection of blocks. Since the computational power is increasing at a rapid speed and the Bitcoin network must keep the block generation time at 10 minutes in average, the target should adjust accordingly. The retargeting is happening dynamically on every full node independently for every 2016 blocks, which occurs every two weeks. The retargeting formula used by Bitcoin full nodes is [118]:

Table 1. DIFFICULTY AND HASH RATE CHANGE BETWEEN 2016 AND 2017[120]

Date	Difficulty	Hash Rate (GH/s)
Dec 6 th , 2017	1,590,896,927,258	11,388,083,790
Dec 2 nd , 2016	286,765,766,821	2,052,749,317
Ratio of change	5.547722606133257	5.547722605818799

Without the difficulty, any miner possessing big hashing power would take over the Blockchain and could change it at will, therefore the difficulty participates strongly to the security of the Bitcoin.

3 Conclusion

All the aforementioned concepts suggest that Bitcoin is a secure by design crypto-currency. Its security relies on the cutting-edge cryptographic technologies such as the digital signature and the hash functions. The Difficulty plays a major role in the Bitcoin Security since it regulates the mining process, so a new block is added to the Blockchain within 10 minutes in average. Also, its dynamic change helps keep up with the increasing hashing rate to avoid Blockchain hijacking by miners with huge computational power. Notwithstanding the difficulty benefits, there is a big issue that Bitcoin community should address, which is the huge electricity consumed by the Miners using their hashing machines to overcome the difficulty.

Finally, to preserve the Bitcoin security, the community should empower the proof-of-work concept while searching for other computing alternatives so that the hashing process would become more energy efficient.

Appendix VIII

Using the randomized solution of the dining philosophers' problem to prevent the Bitcoin majority attack.

B. Lamiri¹, K. Gueraoui², G. Zeggwagh³

Abstract – Bitcoin has been burgeoning since its inception and has become the most successful currency in the crypto-currency market. Despite some unexpected fluctuations in its value, Bitcoin continues to be the most used and spread crypto-currency in the world. This success is due, to some extent, to its security. This security is built over many features for different assets, such as the hashing functions for addresses, the elliptic curve digital signature (ECDSA) for transactions, the elliptic curve discreet logarithm for private keys generation, the difficulty for block mining within a period of 10 minutes, and the proof-of-work as a required condition to add new blocks to the Blockchain. Despite the embedded security features, Bitcoin is still vulnerable to the 51% attack, in which a mining pool gaining most of the network hashing rate could temper with the Blockchain. This paper suggests a way to prevent this issue using the randomized solution of the dining philosophers' problem. This approach would deny to any super-pool the ability to monopolize the mining process if it is implemented in the Bitcoin system. **Copyright © 2010 Praise Worthy Prize Seral. - All rights reserved.**

Keywords: Bitcoin, Difficulty, Dining philosophers' problem, Security, 51% Attack

Nomenclature

AKA	Also Known As
ASIC	Application Specific Integrated Circuit
CTMC	Continuous Time Markov Chains
DDOS	Distributed Denial Of Service
DPP	Dining Philosophers' Problem
DTMC	Discrete Time Markov Chains
ECDSA	Elliptic Curve Digital Signature Algorithm
HEX	Hexadecimal
MDP	Markov Decision Process
PA	Probabilistic Automata
PTA	Probabilistic Time Automata
RIPEMD-160	RACE Integrity Primitives Evaluation Message Digest 160 bits
SHA-256	Secure Hash Algorithm 256 bits
Target_Max	Target Of The Genesis Block (Block#0)
Target_Current	Target Of The Current Block
Time_2016	Time made by 2016 blocks in minutes

I. Introduction

Since its inception in 2009, Bitcoin has been growing at a rapid rate. Its exchange value has reached more than \$20,000 US dollars on December 17th, 2017 before

dropping to \$7145 on May 29th, 2018 for one bitcoin [121]. This makes it the most fluctuating cryptocurrency in the market. Despite this downside, many people are still interested in investing in this crypto-currency and their interest is mainly based on two underlying facts, which are decentralization and security.

Bitcoin is a decentralized system that does not rely on any third party to process the transactions. The transactions are collected and validated by all the participating nodes connected to the peer-to-peer network. The validated transactions are then stored in blocks. These blocks are then added to the Blockchain, which is a distributed ledger that contains all the transactions since the inception of the system.

Bitcoin is considered a secure by design system chiefly because of its embedded cutting-edge cryptographic features, for instance, the Bitcoin keys and addresses generation process is secure because of the use of some practices and technologies, such as the randomness in producing the private keys; the elliptic curve discrete logarithm, which cannot be solved giving the current computational power; the one-way property of the hashing functions SHA-256 and RACE MD (RIPEMD160); and because no backdoors were yet discovered in the elliptic curve. This elliptic curve is defined by a standard known as SECP256K1[122].

This security is enforced by using the Base58Ckeck formatting, which ensures the integrity of the data, mainly

for the Bitcoin keys and addresses. Also, the use of the ECDSA (Elliptic Curve Digital Signature) ensures that only the holders of the private keys can redeem the related funds.

Other security features are added to Bitcoin to ensure the most prominent security objectives such as the confidentiality, the integrity and the availability. These features include, but not limited to, the back-linkage of blocks, which helps ensure the integrity of the Blockchain; the Merkle tree, which ensures the Block integrity; and the difficulty, which ensures the system integrity since it forces miners to work hard for at least 10 minutes to find a proof-of work (POW) for a new block.

Despite these embedded security features, Bitcoin security is still a hot research topic and many security issues and challenges are yet to be handled by Bitcoin community. In this context, Jerry Brito and Peter Van Valkenburgh mentioned six global threats that could harm the functioning of the Bitcoin. These threats are: flawed key generation, transaction malleability, 51% attacks, Sybil attacks, DDOS attacks, and consensus or fork risks [123]. Some of these issues were already addressed by Bitcoin community, such as transaction malleability which was solved by the introduction of the segregated witness in the Bitcoin's Block structure.

The Bitcoin's current big issue is the 51% attacks, aka, the majority attack or the selfish mining. This refers to the ability of a miner or a pool of miners controlling more than half of the network hashing rate [124]. This would grant them on the ability to generate the longest chain in the system. Basically, any malevolent mining pool with more than 51% of the network hashing rate, would be capable of:

- Modifying the transaction data, which may cause double-spending attack [125, 126]
- Preventing confirmations [127]
- Preventing Bitcoin generation [127]

The 51% attack was studied by other researchers who suggested different solutions. Iuon-Chang Lin and Tzu-Chun Liao, in their paper: A Survey Of Blockchain Security Issues and Challenges, 2017 [127], said that the 51% attack was more feasible in the past when most transactions were worth significantly more than the block reward and when the network hash rate was much lower. On the contrary, Martijn Bastiaan in his paper: Preventing the 51%-Attack: A Stochastic Analysis of Two Phase Proof of Work in Bitcoin, said that the 51% attack is still a major security challenge for Bitcoin and suggested a solution based on a two-phase proof of work simulated using a Markov chains model [128].

Ittay Eyal & Emin Sirer in their paper, "Majority is not enough: Bitcoin Mining is Vulnerable", suggest that selfish miners who are detaining more than 33% of the network hashing power can still acquire an important part in the mining process. They mentioned that a selfish mining strategy consists of a miner not announcing his mined blocks to the network in order to increase their

revenue and letting other miners wasting their time and computational power. They suggested a countermeasure to prevent this strategy by urging miners to disseminate all the received blocks and choose randomly one block to mine on it in case of two competing blocks [129].

Arthur Gervais, Hubert Ritzdorf, Ghassan O. Karame, and Srdjan Capkun, in their paper: "Tampering with the Delivery of Blocks and Transactions in Bitcoin", declared that an attacker even with constrained-resources could find a way around the "Eyal & Sirer" security measure by exploiting the Bitcoin object request management system, which would prevent blocks delivery for around 20 minutes. They demonstrated feasibility and easy realization of their attacks in current Bitcoin client through analysis and implementation of some hosts. They showed that the adversary can easily mount Denial-of-Service attacks, considerably increasing his mining advantage in the network or double-spend transactions in spite of the current countermeasures adopted by Bitcoin system. Their contribution consists of a modification of the block request management system in Bitcoin in order to detect any misbehavior in the delivery of blocks and harden the security of the network without deteriorating the scalability of Bitcoin [130].

Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar, in their paper: "optimal selfish mining strategies in Bitcoin", defined a lower threshold of computational power (lower than the one defined by Eyal & Sirer) at which selfish miners could be successful. They cited that attackers with strictly less than 25% of the computational resources can still gain from selfish mining, unlike what Eyal & Sirer conjectured. In addition, they demonstrated how any attacker for which selfish mining is profitable can execute double spending attacks bearing no costs, unlike what the security analysis of Satoshi NAKAMOTO has guessed [131].

Zhenzhen Jiao and Rui Tian and Dezhong Shang and Hui Ding, in their paper: "Bicomp: a bilayer scalable Nakamoto consensus protocol", discussed how Bicomp can resist selfish mining. Their approach is based on high security and pure decentralized Nakamoto consensus, and with a significant improvement on scalability. In Bicomp, two kinds of blocks are generated, i.e., microblocks for concurrent transaction packaging in network, and macroblocks for leadership competition and chain formation. A leader is elected at beginning of each round by using a macroblock header from proof-of-work. An elected leader then receives and packages multiple microblocks mined by different nodes into one macroblock during its tenure, which results in a bilayer block structure. Such design limits a leader's power and encourages as many nodes as possible to participate in the process of packaging transactions, which promotes the sharing nature of the system and resists to selfish mining [132].

In another work entitled, "Resisting Selfish Mining Attacks in the Bicomp", Rui Tian and Wei Gong

mentioned that the selfish mining strategy can comprise a Nakamoto consensus system with less than 25% mining power of the whole system. They have analyzed in detail the selfish mining resistance of the Bicomp protocol. Through modeling the system as a state machine, and analyzing different mining activities that lead state transition, they concluded that the Bicomp can adjust its resistance towards selfish mining attack by varying macroblock POW difficulty and tenure length parameters. They also presented a modification to the Bicomp protocol without substantially modifying the operation mechanism of the system [133].

Jaewon Bae and Hyuk Lim, in their article entitled: “Random Mining Group Selection to Prevent 51% Attacks on Bitcoin”, mentioned that an attacker node whose hash power is greater than half of the total hash power, that node can perform a double-spending attack, i.e., a 51% or majority attack. They proposed an approach to reduce the probability of a successful double-spending attack on Bitcoin. The proposed approach divides miners into groups and gives mining opportunity to a randomly selected group. Their analysis showed that if the number of groups is greater than or equal to two, the probability that the attacker will find the next block is less than 50%. They concluded that this approach can reduce likelihood of a majority attack and can reduce the computing power costs of block mining [134].

Besides the aforementioned mitigation measures, this paper suggests a brand-new way to handle the majority attack in a proactive manner using the randomized solution of the dining philosophers’ problem so that miners even with gigantic hashing rate could not take over the mining process. This article is organized as follow: section II delves into some details about Bitcoin difficulty and how it participates in security, section III provides a quick overview about mining pools, section IV provides some insights about the dining philosophers’ problem and its solutions, section V presents our contribution along with some results and analysis, and section VI provides a conclusion for the paper.

II. Bitcoin Difficulty

Difficulty can be defined as a measure of how difficult it is to find a hash (proof-of-work) below a given target [135]. This parameter is set dynamically by the Bitcoin network every 2016 blocks or two-weeks in average. The difficulty is tied to two other parameters, the target and the Bits, which we will explain in the following paragraphs.

The proof-of-work serves as a proof that the miner has committed a great amount of hashing power to find the block header’s hash that satisfies the required condition. The proof-of-work is hard to find but easy to verify. It involves finding a value for the nonce that results in a block’s header hash, using SHA-256 algorithm, that is less

or equal to the difficulty target, AKA target.

Every block contains a field called “Bits”, known also as target Bits, which is a four-byte number represented in a hexadecimal floating-point format. Bits value serves to calculate the difficulty target, which is used as a condition in the mining algorithm. The Bits field value of the first block in the Blockchain is 1d00ffff. By convention, the first two digits (1d) represent the total number of digits a target is made of. It is used in the exponent of the floating-point notation while the remaining digits (00ffff) represents the coefficient.

To calculate the target from the Bits value, we rely on the following formula:

$$\text{TARGET} = \text{COEFFICIENT} * 2^{(8 * (\text{EXPONENT} - 3))} \quad (1)$$

Using the hexadecimal representation and applying this formula to the block #0 with Bits value of (0x1d00ffff), the target would be:

$$\text{TARGET} = 0x00FFFF * 2^{(8 * (0x1d - 0x3))}$$

Therefore, the result in hexadecimal format is:

$$\text{TARGET (in HEX)} = 0xffff00$$

The target condition sets the frequency at which a new proof-of-work is found. It determines also the difficulty for a collection of blocks. Since the computational power is increasing at a rapid speed and the Bitcoin network must keep the block generation time at 10 minutes in average, the target should adjust accordingly. The retargeting is happening dynamically on every full node independently for every 2016 blocks, which occurs every two weeks. The retargeting formula used by Bitcoin full nodes is:

$$\text{NEW TARGET} = \frac{\text{TARGET_CURRENT} * (\text{TIME_2016})}{20160 \text{ MINUTES}} \quad (2)$$

The difficulty is tightly linked to the target and shows how it is difficult to find a new hash of a block that satisfies the target condition. Its main purpose is to regulate the mining process, so a new block is mined every 10 minutes in average. It is calculated using the following formula:

$$\text{DIFFICULTY} = \frac{\text{TARGET_MAX}}{\text{TARGET_CURRENT}} \quad (3)$$

The difficulty is tightly linked to the hashing rate. When the hashing rate increases, the proof-of-work is found quickly and therefore the difficulty increases too to keep the proof-of-work finding around 10-minutes in average. Also, when proof-of-work discovery time is slower, the difficulty decreases. Table I illustrates the strong correlation between the difficulty and the hashing rate. It shows also that difficulty and the hash rates have quintupled since the last year. This is due mainly to

competition between miners.

TABLE I
DIFFICULTY AND HASHRATE CHANGE BETWEEN 2016 AND 2017[1].

Date	Difficulty	Hash Rate (GH/s)
Dec 6 th , 2017	1,590,896,927,258	11,388,083,790
Dec 2 nd , 2016	286,765,766,821	2,052,749,317
Ratio of change	5.547722606133257	5.547722605818799

Without the difficulty, any miner possessing big hashing power could tamper with the system and may cause some unwanted changes, therefore the difficulty participates strongly to the security of the Bitcoin. However, mining equipment consumes a huge amount of electricity, so to keep up with the ever-increasing difficulty. Thus, miners are joining pools to optimize their benefit.

III. Bitcoin Mining Pools

Another security feature of Bitcoin lays on the tremendous work that the miners that are doing to solve the POW, which plays as a guarantee of the Blockchain integrity. Miners are continuously in race, with huge warehouses full of dedicated hashing machines called ASIC (Application Specific Integrated Circuit). This gave rise to the formation of mining pools, in which individual miners pool their processing power to increase their probability of winning [137]. The winning pool share the reward among the participants according to their dedicated hashing rate. Currently, there are 19 mining pools in the Bitcoin mining business. Fig. 1 illustrates these mining pools along with the percentage of their hashing rate in the whole network.

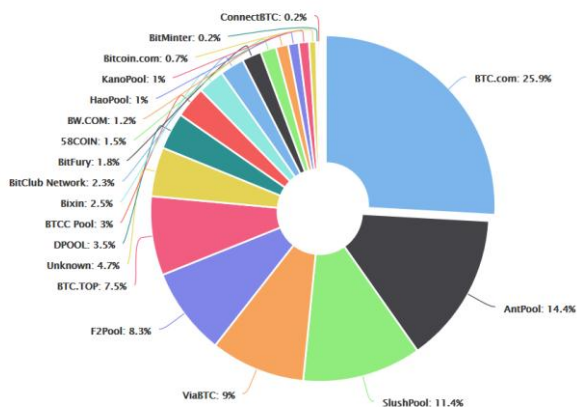


Fig.1. Bitcoin mining pools and their hash-rate distribution [138].

The five major Bitcoin mining pools are: BTC.com, AntPool, SlushPool, ViaBTC, and F2Pool. These pools hold 69% of the entire network hashing rate. Currently, no mining pool is detaining more than 50% of the computational power, however in July 2014, Ghash.io has gained more than half of the network hashing capacity for an extended period. Luckily, they publicly promised to not attack the system to avoid damaging confidence in Bitcoin [139].

Despite the benefits of pooling mining equipment, some security issues could arise when pools attain more than half of the network hashing power. This issue is known as the 51% attack challenge. More details are provided in the next section.

IV. Dining Philosophers Problem

The dining philosophers problem (DPP) is a classic synchronization problem which is used to evaluate situations where there is a need of allocating multiple resources to multiple processes [140].

There are N numbers of philosophers sitting around a round table eating noodles and sharing ideas as well as thinking. Each philosopher requires two chopsticks to eat, and there is one chopstick between two philosophers. The purpose is that no one starves, and maximum number of philosophers can eat at the same point of time [141]. At any moment, a philosopher is either eating or thinking.

In the dining philosophers' problem, two neighbors cannot eat at the same time. The maximum number of philosophers eating at the same time is equal to the integer part of N/2, where N is the number of philosophers. There are three states for each philosopher: hungry, eating, and thinking. Fig. 2 shows 8 dining philosophers sitting around a table.

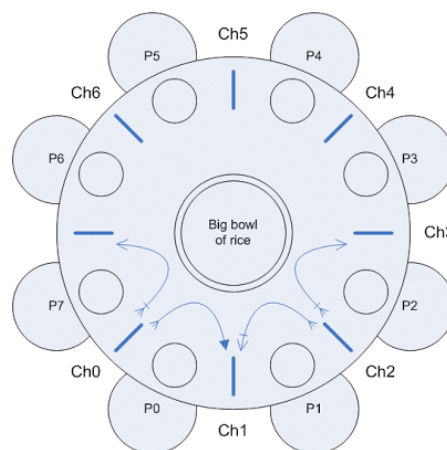


Fig. 2. Eight sitting dining philosophers [142]

A solution to this dining philosopher problem is represented in Fig. 3:

```
void Philosopher (int i)
{
    while(True)
    { THINK;
      PICKUP(CHOPSTICK[i]);
```

```

PICKUP(CHOPSTICK[i+1 mod N]);
EAT;
PUTDOWN(CHOPSTICK[i+1 mod N]);
PUTDOWN(CHOPSTICK[i]);
}
}
    
```

Fig.3. A basic solution of the DPP

This code solves the DPP, but it could lead to a deadlock, in which no philosophers is able to eat because one of his required chopstick would be taken by another philosopher. This happens when philosophers take one chopstick at the same time, only the left or only the right, as shown in the Fig. 4.

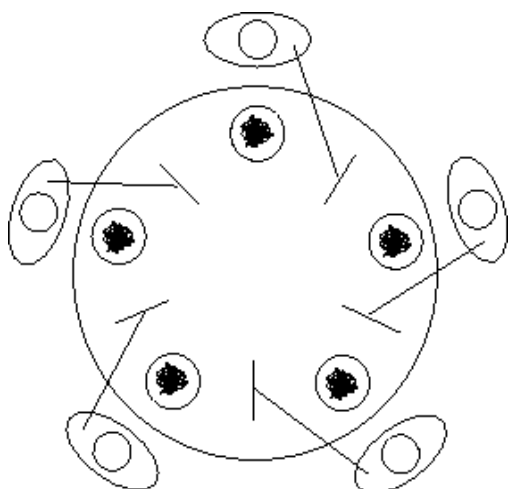


Fig.4. A deadlock in the DPP [143].

The easiest way to avoid the deadlock is to make sure that at least one philosopher is taking his first chopstick from the right side while the others are taking their first chopstick from the left side or vice versa. Fig. 5 illustrates this solution [144]:

```

void Philosopher(int i)
{
while(True)
{ THINK;
PICKUP(CHOPSTICK[min(i, i+1 mod N)]);
PICKUP(CHOPSTICK[max(i, i+1 mod N)]);
EAT;
PUTDOWN(CHOPSTICK[min(i, i+1 mod N)]);
PUTDOWN(CHOPSTICK[max(i, i+1 mod N)]);
}
}
    
```

Fig.5. A solution to the deadlock problem in the DPP

Another solution to the deadlock problem is to introduce a central entity, as an arbitrator, who gives permission to only one philosopher at a time and make sure that the philosopher has picked up both of his chopsticks. This solution, though efficient, adds a little of centralization in the system. Other solutions to the same problem are available, such as Chandy/Misra solution [145] and Dijkstra's solution [146].

Besides the deadlock, another problem, known as starvation may occur. This happens when one or more philosophers will not be able to eat at all because other philosophers may monopolize the chopsticks. Fig. 6

depicts this issue, where Philosophers A&C, B&E can take turns to pick the chopsticks to such a degree that philosopher D starves out.

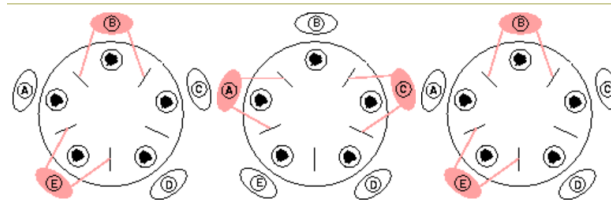


Fig.6. Starvation issue in the DPP [19].

A good solution of the DPP should not have any deadlock or starvation. In this context, Lehmann and Rabin solved the resource-starvation problem in the following way:

A philosopher will not pick up his/her neighbor's chopstick (when it has no chopsticks) if that neighbor is trying to eat and has not eaten since the philosopher's most recent meal [147].

In the next section, we will rely on this implementation to provide an alternative way to prevent the 51% attack in Bitcoin.

V. Contribution

In this paper, we are suggesting the randomized solution of the dining philosophers' problem to prevent the Bitcoin 51% attack. Before being able to race for a new POW, mining pools should get their chopsticks through an arbitrator node, which should organize this process to avoid the deadlock or the starvation problems. Fig. 7 illustrates this process for five mining pool.

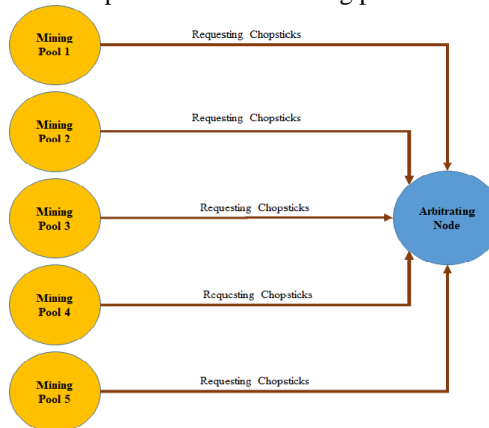


Fig.7. Arbitrating node in Bitcoin

The arbitrating node that we are suggesting will make sure that every mining pool will get a chance to mine and no one will monopolize the mining process. The mining process will be organized in two phases: picking the chopsticks and mining. While the first phase adds some centralization in the system, the second one will still be completely decentralized.

For this purpose, we adapted the Lehmann and Rabin solution to the mining process by:

- Replacing Philosophers with mining pools.
- Replacing the three states of philosophers (Hungry, Eating, Thinking) with three states for miners (Hungry to mine, Mining, resting to cool down equipment and stop consuming electricity).

Picking two chopsticks (right and left) is a required condition to start the mining race. This condition will deny to any mining pool, regardless of its hashing capacity, a hijacking of the mining process and will give all mining pool a fair chance to win a reward in the system.

Currently, there are 19 mining pools in the system. With this implementation, only 9 pools (the integer part of 19/2) will be competing at a time for a POW.

This paper relies on Markov Decision Process (MDP), to simulate the dining philosophers' problem solution based on the Lehmann and Rabin's randomized solution.

MDP is widely used by the research community, which allows us to take advantage of previous works and the available literature. To build the model, we relied on a model checker software called Prism.

Prism provides analysis for systems that shows a probabilistic behavior. It is used for different research areas, such as communication and multimedia protocols, stochastic algorithms, security protocols, and biological systems. PRISM can build and analyze several types of probabilistic models [148]:

- discrete-time Markov chains (DTMCs)
- continuous-time Markov chains (CTMCs)
- Markov decision processes (MDPs)
- probabilistic automata (PAs)
- probabilistic timed automata (PTAs)

In this study, we are using Markov decision process to analyze the solution we are suggesting for the 51% attack. The following diagram (see Fig. 8) illustrates the 12 states of the process and the actions taken between two states.

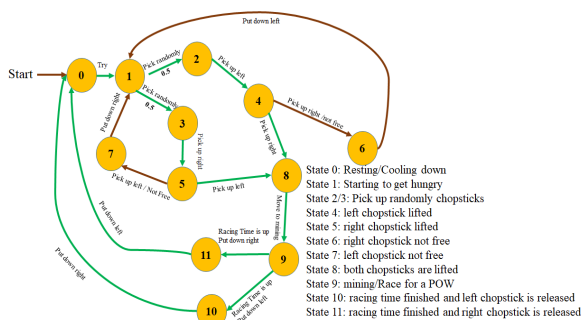


Fig. 8. State diagram of the process.

We split the three states of a mining pool into twelve states to see specific details of the whole process. This model is implemented in prism for three miners as illustrated in the following code as shown in Fig. 9 [149]:

```
// left Chopstick free and right Chopstick free resp.
// left neighbor is Miner 2
formula lfree = (M2>=0&M2<=4)|M2=6|M2=10;
// right neighbor is Miner 3
formula rfree = (M3>=0&M3<=3)|M3=5|M3=7|M3=11;
module Miner1
  //12 state-diagram
  M1: [0..11];
  [] M1=0 -> (M1'=1); // trying
  [] M1=1 -> 0.5 : (M1'=2) + 0.5 : (M1'=3); // pick randomly
  [] M1=2 & lfree -> (M1'=4); // pick up left chopstick
  [] M1=3 & rfree -> (M1'=5); // pick up right chopstick
  [] M1=4 & rfree -> (M1'=8); // pick up right chopstick (got
left)
  [] M1=4 & !lfree -> (M1'=6); // right chopstick not free (got
left)
  [] M1=5 & lfree -> (M1'=8); // pick up left chopstick (got
right)
  [] M1=5 & !lfree -> (M1'=7); // left chopstick not free (got
right)
  [] M1=6 -> (M1'=1); // put down left chopstick
  [] M1=7 -> (M1'=1); // put down right chopstick
  [] M1=8 -> (M1'=9); // move to Mining (got both chopsticks)
  [] M1=9 -> (M1'=10); // racing time finished and left
chopstick is released
  [] M1=9 -> (M1'=11); // racing time finished and right
chopstick is released
  [] M1=10 -> (M1'=0); // put down right chopstick and return
to resting
  [] M1=11 -> (M1'=0); // put down left chopstick and return to
resting
endmodule
// construct further modules through renaming
module Miner2 = Miner1 [ M1=M2, M2=M3, M3=M1 ] endmodule
module Miner3 = Miner1 [ M1=M3, M2=M1, M3=M2 ] endmodule
// rewards (number of steps)
rewards "num_steps"
  [] true : 1;
endrewards
// labels
label
  label "Hungry" =
((M1>0)&(M1<8))|((M2>0)&(M2<8))|((M3>0)&(M3<8));
  label "Mining" =
((M1>=8)&(M1<=9))|((M2>=8)&(M2<=9))|((M3>=8)&(M3<=9));
```

Fig. 9. Randomized solution of the DPP adapted to Bitcoin Mining Process

We relied on the following model checking, which is depicted in Fig. 10, to check the number of iteration and the time it takes for each miner get a chance to race for a POW; we checked also the number of iterations before the first miner get a chance to mine.

```
const int K; // discrete time bound
// liveness (if a Miner is hungry then eventually some Miners race for
a POW)
"Hungry" => P>=1 [ true U "Mining" ]
// bounded waiting (minimum probability, from a state where
someone is hungry, that a Miner will mine within K steps)
Pmin=?[true U<=K "Mining" {"Hungry"} {min}]
// expected time (from a state where someone is hungry the maximum
expected number of steps until a Miner mines)
Rmax=?[F "Mining" {"Hungry"} {max}]
```

Fig.10: Model checking.

The results below illustrate some statistics for the MDPs we have built for different values of the constants N=3 (number of miners) and K (number of iterations).

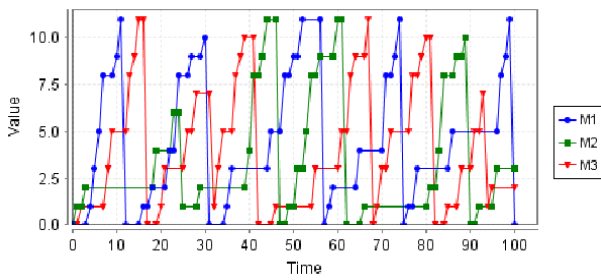


Fig. 11. Miners different states (0 to 11) for 100 iterations

Fig.11 shows that miner 1 start mining after 10 iterations, while miner 2 and 3 got their chances after 14 and 44 iterations respectively.

Fig.12 depicts that the minimum time required for a miner to start mining is around 0.0625 seconds and the maximum iterations needed for all miners to get a chance to mine is about 50.

Property: "Hungry"=>P>=1 [true U "Mining"]
Defined constants: <none>
Method: Verification
Result: true (property satisfied in the initial state)

Property Details

Property: Pmin=? [true U<=K "Mining" ("Hungry"){min}]
Defined constants: <none>
Method: Verification
Result: 0.0625 (minimum value over states satisfying filter)

Property Details

Property: Rmax=? [F "Mining" ("Hungry"){max}]
Defined constants: <none>
Method: Verification
Result: 50.99852574264277 (maximum value over states satisfying filter)

Fig.12. Results of the model checking properties.

The results change as we change the number of miners. For more analysis and details, we tested one implementation of the randomized solution of the dining philosophers adapted to the mining process. The following paragraphs provide more detail.

For testing purposes, we used 19 mining pools, numbered from 1 to 19 and an adaptation to a solution of

the DPP, written in Java language, to simulate the role of the Arbitrating Node [150]. Fig. 12 shows an excerpt of the used code and some of the results.

```

while (true) {
    // Resting
    doAction(System.nanoTime() + ": Resting");
    synchronized (leftChopstick) {
        doAction(
            System.nanoTime()
            + ": Picked up left Chopstick");
        synchronized (rightChopstick) {
            // eating
            doAction(
                System.nanoTime()
                + ": Picked up right Chopstick - Mining");

            doAction(
                System.nanoTime()
                + ": Put down right Chopstick");
        }
    }
}

```

Code

```

Mining Pool:8 309921003803648: Resting
Mining Pool:7 309921003843549: Picked up right Chopstick - Mining
Mining Pool:15 309921034319353: Put down left Chopstick. Back to Resting
Mining Pool:16 309921034334760: Picked up left Chopstick
Mining Pool:13 309921044387485: Resting
Mining Pool:12 309921044392226: Picked up right Chopstick - Mining
Mining Pool:7 309921054014729: Put down right Chopstick
Mining Pool:18 309921066046710: Resting
Mining Pool:17 309921066089377: Picked up right Chopstick - Mining

```

Results

Fig. 12. Excerpt of code and results used for testing

After running this code several times, we realized that all the mining pools got a chance to mine and eventually add a new block in the Blockchain independently of their hashing rate (see Fig. 13). The results show that mining pools get their chopsticks randomly. This denies to miners the ability to mine many blocks successively, which is a condition that is difficult to reverse in Bitcoin.

14 16 8 18 11 6 13 10 15 17 18 5 12 9 14 16 4 13 11
 15 8 17 10 3 14 16 18 9 12 7 2 17 11 8 13 1 6 18 10 16
 12 9 19 5 7 15

Fig. 13. Randomized order of mining for the 19 mining pools different for each attempt.

Giving all these results, we concluded that the randomized solution of the DPP implemented within an arbitrating node would prevent the monopoly of the mining process for any super-pool, holding a hashing rate capacity of more than 50%.

VI. CONCLUSION

Bitcoin is a secure by design crypto-currency that relies on cutting-edge cryptographic technologies such as the digital signature and the hash functions. In addition, the Difficulty plays a major role in the Bitcoin Security since it regulates the mining process, so a new block is added to the Blockchain within 10 minutes in average. Despite all these features, Bitcoin is still vulnerable to 51% attacks. This paper provided a new way to regulate the mining process, so no pool will hijack the system by using the randomized solution of the dining philosophers' problem, which should be implemented in an arbitrating node. This study provided also some analysis of the different states of mining pools using a Markov Decision Process model, implemented in Prism. The results showed that the suggested solution works perfectly to organize the mining process and grant each mining pool a chance in the POW race and prevent monopoly of super-pools. Therefore, we are strongly recommending Bitcoin community to consider this alternative as a way to prevent the 51% attack.

Finally, Bitcoin community should consider this solution while working on ways to make it decentralized or yield for some centralization for security purposes.

Appendix IX

Bitcoin embedded security items review and center of gravity analysis

Abdenaby Lamiri
Research Center on Energy
Mohamed 5th University
Rabat, Morocco
abdlamsic@gmail.com

Kamal Gueraoui
Research Center on Energy
Mohamed 5th University
Rabat, Morocco
kgueraoui@yahoo.fr

Gamal Zeggwagh
Research Center on Energy
Mohamed 5th University
Rabat, Morocco
gamalzeggwagh972@hotmail.com

Abstract— Bitcoin became the most prominent cryptocurrency and payment system in the market since its inception in 2009. A decentralized system that allows users to send and receive transactions without a need for a third party to process them. Its security has been continuously improved thanks to the academics and research community. This paper provides an overview on some key embedded security features and a brief analysis using the critical factor analysis framework to determine its critical capabilities, its critical requirements, its critical vulnerabilities, and its center of gravity along with some strategies to disrupt or prevent disruption of the system.

Keywords— Analysis, Bitcoin, Center of gravity, Crypto-Currency, Security.

I. INTRODUCTION

Bitcoin was forged by SATOSHI NAKAMOTO after years of research for a decentralized cryptocurrency. It took advantage of the cutting-edge cryptographic constructs, such as hash functions and the elliptic curve digital signature algorithm. For the first time in history, online payment and money exchange could be made in a decentralized way.

Bitcoin works over a peer-to-peer networks of nodes playing different roles, such as mining nodes, full nodes, lightweight nodes, and peer-to-peer overlay network. It relies on specific data structures such as transactions, blocks, and the Blockchain. All the Blockchain data are made public, which makes Bitcoin an open system to the public without concerns about confidentiality, however Bitcoin promised some privacy for its users. In addition, Bitcoin data integrity is mostly ensured by hashing functions, Merkle tree, and back-linkage of blocks. On the other hand, its availability is based primarily on the mining process and the peer-to-peer network. Bitcoin security is a hot research topic which interests many researchers and academic around the world.

It must be said that Bitcoin is not only a cryptocurrency, but a methodology giving rise now to the so-called Blockchain technology, finding many applications in diverse areas such as economy, financial problems, smart contracts, etc. There is many excellent surveys on Bitcoin treating issues in depth and comprehensive manner such as [151, 152, 153]. As an example, Ron and Shamir [154] introduced heuristic techniques to analyze privacy based on graph analysis showing that there is no absolute anonymity. This work was followed by [155, 156] using non-interactive zero-knowledge proofs to enforce privacy resulting in new altcoins. The work of [157] proposes other techniques to structure Blockchains with relations to Ethereum [158] modifying parts of the original Bitcoin protocol. Fundamental works such as [159, 160] analyze the NAKAMOTO protocol with relations to the well-known Byzantine consensus. As noticed by many researcher Bitcoin must be studied on practical as well as theoretical foundations. This paper aims to determine the Bitcoin center of gravity to increase awareness on how the system might be attacked or defended. It is organized as follow: Section II lays out the major embedded security features of Bitcoin, Section III provides a quick analysis on Bitcoin Center of Gravity and ways to suppress and protect Bitcoin, and Section IV provides a conclusion for the Paper.

II. BITCOIN EMBEDDED SECURITY FEATURES

In this section, we will go through the most important security features that are embedded within the Bitcoin system. There features are brought by the cutting-edge innovations in the realms of cryptography and distributed systems. They include hashing functions, elliptic curve, digital signature, and encodings. The analysis is done based on three security objectives, which are Confidentiality, Integrity, and Availability, also known as CIA.

A. Bitcoin Keys and Addresses Security

A Private Key, which serves as a proof-of-ownership, is a 256-bit number picked randomly using the operating system entropy (randomness). It is derived from random function or a hash of some data. There is 2256 numbers of possibilities, which makes it hard to predict. In addition, private keys are stored in an encrypted format using a symmetric encryption system, which is the advanced encryption system (AES).

TABLE I: PRIVATE KEY SECURITY IN TERMS OF C.I.A

	Confidentiality	Integrity	Availability
Private Key	- Depends on the level of randomness - Tied also to the AES encryption System including the key management.	- Ensured by the WIF encoding - The AES encryption system.	Tied to the availability of the Bitcoin wallet

A Public Key, which represents a point in the Elliptic Curve, defined by secp256k1 Standard [161], is related to the private key. It is derived using a discrete logarithm formula (See Equation 1), is proven to be hard to break giving the current

B. Lamiri, K. Gueraoui, G. Zeggwagh

computational power. Public key security relies on the unbreakable the elliptic curve discrete logarithm (ECDL) formula, where G is the Generator point of the Koblitz elliptic curve (Equation 2)

$$PUBKEY = PRIVKEY * G \quad (1)$$

$$EC: Y^2 = X^3 + 7 \quad (2)$$

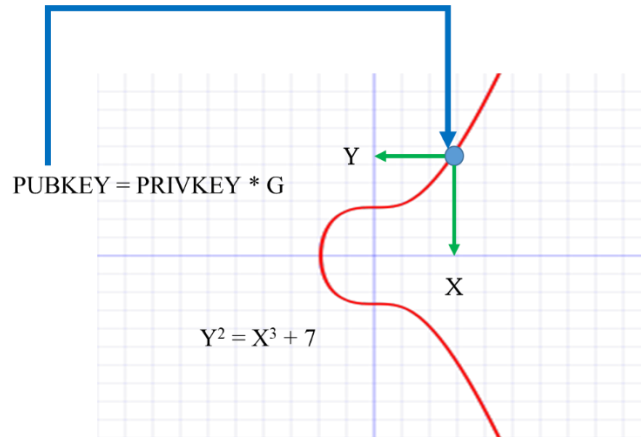


Fig.1. Secp256k1 defined elliptic curve.

TABLE II. PUBLIC KEY SECURITY IN TERMS OF C.I.A

	Confidentiality	Integrity	Availability
Public Key	Relies on the discrete logarithm formula and the no-backdoor in the elliptic curve until now	Ensured by Base58Check encoding	Tied to the availability of the Bitcoin wallet

Bitcoin Address, identifies users and serves to send and receive funds from one user to another. It is derived from the public key using hash-functions, such as SHA-256 and RipeMD-160 (See Equation 3).

$$ADDRESS = RIPEMD160 (SHA256 (PUBKEY)) \quad (3)$$

TABLE III. BITCOIN ADDRESS SECURITY IN TERMS OF C.I.A

	Confidentiality	Integrity	Availability
Bitcoin Address	Relies on the no-collision in the hashing functions: SHA-256 and RIPEMD-160	Ensured by Base58Check encoding	Tied to the availability of the Bitcoin wallet

Overall, Bitcoin keys and addresses security depends on:

- The randomness used in producing the private key
- The elliptic curve discrete logarithm, which can't be solved giving the current computational power
- The one-wayness property of the hashing functions SHA-256 and RACE MD (RIPEMD160)
- And also because no backdoors were discovered in the elliptic curve till now

If a collision is to happen, in the future, in SHA-256 function, Bitcoin community should be prepared to shift to SHA-512 in order to prevent producing the same address for different private keys.

B. Bitcoin Wallet Security

Wallets are applications used to send/receive transactions, to track user's balances, and to store user's private/public keys. They run on different platforms, such as windows, Linux, MacOS, etc. They can be also in a hardware and paper format. Hardware wallets use electronic devices that store private keys and are considered the most secure of all kinds of wallets. Wallets security depends more on the user's awareness of the threats and risks that are related to private keys security, such as encryption. Bitcoin keys are encrypted, with a master key which is entirely random, using the Advanced Encryption Algorithm (AES) [162].

B. Lamiri, K. Gueraoui, G. Zeggwagh

C. Bitcoin Scripting Language

Bitcoin uses a stack-based language with simple and limited functions, known as opcodes. It supports functions that serve for comparison, hashing, and signature verification. These functions are mainly used to lock and unlock transactions. For security purposes, loop functions are disabled, which deny to any attacker the possibility of crafting denial of service attacks [163].

D. Bitcoin Transaction security

Bitcoin Transactions consist of inputs and outputs that define the sender and the receiver of the funds. They are secured using the Elliptic Curve Digital Signature (ECDSA) to ensure that funds can only be spent by their rightful owners [164]. This signature lies heavily on the elliptic curve security. If a backdoor is found in the secp256k1 elliptic curve, this signature will be vulnerable to brute force attacks. Therefore, Bitcoin transactions are secure as far as the elliptic curve digital signature is still secure.

E. Bitcoin Blocks Security

Blocks are created by miners who succeed in finding the proper proof of work (POW). Their structure is based on a header and a set of valid transactions. The header contains a hash of the previous block, which plays a major role in the block integrity. Blocks are broadcast publicly to the connected nodes in the network. The integrity of the enclosed transactions is ensured by the Merkle tree, which provides a hash of the included transactions.

F. Bitcoin Blockchain Security

In Bitcoin, each block is linked to the previous one. These chain of blocks is what makes the Blockchain. It is also made public in the network. The back-linkage of blocks helps ensure some security for the Blockchain. If a block is altered, all the following blocks should be altered and their POW puzzle should be resolved, which is time and resources consuming process. It is computationally impossible to change a block after it has been confirmed by six other blocks [165]. Sometimes, a fork in Blockchain happens when two blocks are found at the same time by two different miners. Bitcoin solve this issue by considering the longest chain with the largest difficulty as the valid version of history.

G. Mining process Security

Miners get reward for each mined block. This incentive works to keep miners working for the system and not against it. The reward is currently 12.5 BTC and this help them offset the cost of consumed electricity.

The difficulty regulates the block production process, so a new block is added to the Blockchain every 10 minutes in average. This feature helps Bitcoin system to adapt to the continuously increasing hashing rate.

III. CENTER OF GRAVITY ANALYSIS

In this section, we define the center of gravity for any given system and we provide a quick analysis of critical factors related to Bitcoin.

A. Definitions

According to US Army JP5-0, a COG is a source of power that provides moral or physical strength, freedom of action and the will to act. A COG is analyzed within a framework of three critical factors, which are: Critical capabilities (CC), Critical requirements (CR), and Critical vulnerabilities (CV)[166].

- CC is the primary ability, it is what the COG is able to do;
- CR is essential conditions, resources, or means required by which the COG performs its CC;
- CV are CRs that are deficient or vulnerable; they may be transient and internal or external.

B. Critical factors analysis

The following table provides an overview of the conducted analysis of the critical factors and a suggestion of a center of gravity for Bitcoin at two different levels: the strategic one which depicts high-level goals, and the operational one which represents the daily, weekly, monthly actions that should be done for the survival of the system.

TABLE IV: AN OVERVIEW OF THE BITCOIN CRITICAL FACTORS ANALYSIS

<p>COG:</p> <ul style="list-style-type: none"> • Bitcoin Strategic COG : is the trust that has gained within its users ; • Bitcoin operational COG: the consensus mechanism (POW) that allow the system to confirm TXs, Create new Block, and generate brand new bitcoins. 	<p>CC:</p> <ul style="list-style-type: none"> • create new units of currency (bitcoins created through the mining process) • send/receive bitcoins (use of Bitcoin as a payment system)
<p>CV:</p> <ul style="list-style-type: none"> • Vulnerability to the majority attacks • Loss of private keys accidentally or in case of the death of the owner • Denial of service attacks • Blockchain forks • Transactions latency • Throughput limitation • Energy consumption 	<p>CR:</p> <ul style="list-style-type: none"> • Bitcoin P2P NTW • Wallet applications • Mining process • Internet • Profitability • Strength of Digital signature algorithm • Strength of keys generation

C. Bitcoin disruption strategies

Malicious attackers who seek to disrupt Bitcoin can either opt for a direct approach or an indirect one:

- A direct approach would target directly Bitcoin COG by influencing users’ judgment through lies and propaganda
- An indirect approach would target the Bitcoin CR, which would deny to Bitcoin its CC and therefore lose its brand image and thus lose the trust of its users. This could be achieved through crafting bugs and issues within these CRs. Malicious actions such as owning more than 50% of the hashing power to suppress the consensus mechanism would have a serious impact on transactions confirmation, blocks creation and new currency issuance.

Both a direct and indirect approaches would accomplish the same malevolent purpose, which is the disruption of the system. A best way to secure Bitcoin is to prevent malicious users from exploiting Bitcoin CVs through continuous monitoring and proactive fixes that would strengthen the CRs and protect the COG at both levels.

IV. CONCLUSION

This paper provided an overview of some embedded security features within the Bitcoin ecosystem such as randomness in producing private keys, the elliptic curve discreet logarithm, unbreakable until today, which help produce public keys; the ECDSA that help ensure that TXs are redeemable only by the holders of the private keys; the unbroken properties for the hashing function SHA-256, which are one-wayness and collision-resistance; the denial of service resistance of the Bitcoin scripting language; the Merkle tree and the back-linkage that help ensure the integrity of the blocks and the Blockchain; the reward of miners which make them work for the system and the difficulty that regulates the mining process according to the network hashing power. The analysis of the different factors showed that bitcoin center of gravity is the trust that has gained among its users and the consensus mechanism based on the POW that makes the system working properly.

The paper provided also two main strategies to disrupt bitcoin by influencing the judgement of its users as a direct approach and another way that target its critical requirements mainly the mining process which help maintain the system working properly. The paper suggested continuous monitoring and fixes in a proactive manner to strengthen Bitcoin critical requirements. Since many Bitcoin functions are based on this function, such as POW, Merkle Tree, TX id, any collision could disrupt the functioning of the system. Hence, Bitcoin community should always monitor the strength of this function and prepare a way ahead to prevent any disruption in case of a collision in SHA-256. Implementation of SHA-512 in the system could serve as alternative.

B. Lamiri, K. Gueraoui, G. Zeggwagh

REFERENCES

REFERENCES

- ¹ Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.
- 2 Coinmarketcap, <https://coinmarketcap.com/>, accessed on October 6th, 2018.
- 3 Coinmarketcap, <https://coinmarketcap.com/>, accessed on October 6th, 2018.
- 4 Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.
- 5 Ittay Eyal, Emin Gün Sirer: Majority is not enough: bitcoin mining is vulnerable. *Commun. ACM* 61(7): 95-102 (2018).
- 6 A. Gervais, H. Ritzdorf, G.O. Karame, and Srdjan Capkun in their paper, “Tampering with the Delivery of Blocks and Transactions in Bitcoin”, 2015.
- 7 A.Sapirshstein, Y. Sompolinsky, and A. Zohar, «Optimal selfish mining strategies in bitcoin», 2015.
- 8 Nicolas T. Courtois, Lear bahack: «On Subversive Miner Strategies and Block Withholding Attack in Bitcoin Digital Currency», 2014.
- 9 Ittay Eyal, The Miner’s Dilemma, 2015.
- 10 Zhenzhen Jiao, Rui Tian, Dezhong Shang and Hui Ding. Bicomp: A Bilayer Scalable Nakamoto Consensus Protocol, *CoRR*, 2018, <https://arxiv.org/abs/1809.01593>.
- 11 Rui Tian, Wei Gong: Resisting Selfish Mining Attacks in the Bicomp. *CoRR* abs/1809.06289 (2018).
- 12 Jaewon Bae, Hyuk Lim: Random Mining Group Selection to Prevent 51% Attacks on Bitcoin. *DSN Workshops 2018*: 81-82.
- 13 C. Decker and R. Wattenhofer, “Bitcoin Transaction Malleability and MtGox”, 2014.
- 14 M.Andrychowicz, S. Dziembowski, D.Malinowski, and L. Mazurek, “On the Malleability of Bitcoin Transactions”, 2015.
- 15 Kiran, M and Stannett, M. *Bitcoin Risk Analysis*. <http://www.nemode.ac.uk/wp-content/uploads/2015/02/2015-Bit-Coin-risk-analysis.pdf>.
- 16 Brito, J and Van Valkenburgh, P. Bitcoin: Risk Factors For Insurance. <https://www.lloyds.com/news-and-insight/risk-insight/library/technology/bitcoin>.
- 17 Bitcoin and Cryptocurrency Technologies, <https://fr.coursera.org/learn/cryptocurrency>, accessed on October 14th, 2018.
- 18 Merkle Tree, <https://brilliant.org/wiki/merkle-tree/>, accessed on October 15th, 2018.
- 19 ECDSA, https://en.bitcoinwiki.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm, accessed on October 15th, 2018.
- 20 Private Key. Source: https://en.bitcoin.it/coinmarketcap/wiki/Private_key. Accessed on June 5th, 2017.
- 21 Private Key generation. <https://www.bitaddress.org/bitaddress.org-v3.3.0-SHA256-dec17c07685e1870960903d8f58090475b25af946fe95a734f88408cef4aa194.html>. Accessed on June 6th, 2017.
- 22 Certicom Research. Standards for Efficient Cryptography. SEC 2: Recommended Elliptic Curve Domain Parameters. (n.d.). Retrieved from <http://www.secg.org/sec2-v2.pdf>, accessed on June 5th, 2017.
- 23 Secp256k1. <https://en.bitcoin.it/wiki/Secp256k1>. Accessed on June 6th, 2017.
- 24 Source: <https://en.bitcoin.it/wiki/Secp256k1>. Accessed on June 5th, 2017
- 25 Source: <http://grauai.de/code/elliptic2/>. Accessed on June 6th, 2017.
- 26 Andreas Antonopoulos. *Mastering Bitcoin*. 2nd Edition, 2017.
- 27 Andreas Antonopoulos. *Mastering Bitcoin*. 2nd Edition, 2017.
- 28 Andreas Antonopoulos. *Mastering Bitcoin*. 2nd Edition, 2017.
- 29 Andreas Antonopoulos. *Mastering Bitcoin*. 2nd Edition, 2017.
- 30 Compressed and uncompressed private and public keys. <https://esotera.eu/btc-addresses/>. Accessed on June 8th, 2017.
- 31 Andreas Antonopoulos. *Mastering Bitcoin*. 2nd Edition, 2017.
- 32 P2SH address. https://en.bitcoin.it/wiki/Pay_to_script_hash, on June 6th, 2017.
- 33 Bitcoin wallets. <https://99bitcoins.com/bitcoin-fueling-future-platforms-1>. Accessed on June 1st, 2017.
- 34 Andreas Antonopoulos. *Mastering Bitcoin*. 2nd Edition, 2017.
- 35 Andreas Antonopoulos. *Mastering Bitcoin*. 2nd Edition, 2017.
- 36 CDK functions. <https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki>
- 37 Andreas Antonopoulos. *Mastering Bitcoin*. 2nd Edition, 2017.
- 38 Bitcoin Paper wallet. www.bitaddress.org, on June 8th, 2017.
- 39 Bitcoin Paper wallet. www.bitaddress.org, on June 8th, 2017.
- 40 Number of Unspent Transaction Outputs. <https://blockchain.info/charts/utxo-count>, accessed on June 1st, 2017.
- 41 Andreas Antonopoulos. *Mastering Bitcoin*. 2nd Edition, 2017.
- 42 Andreas Antonopoulos. *Mastering Bitcoin*. 2nd Edition, 2017.
- 43 Predicting bitcoin fees for transactions. <http://bitcoinfoees.21.co/>, accessed on June 1st, 2017.
- 44 Predicting bitcoin fees for transactions. <http://bitcoinfoees.21.co/>, accessed on June 10th, 2017.
- 45 Andreas Antonopoulos. *Mastering Bitcoin*. 2nd Edition, 2017.
- 46 Andreas Antonopoulos. *Mastering Bitcoin*. 2nd Edition, 2017.
- 47 Common vulnerabilities and exposures. Cve-2012-3789. <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-3789>, accessed on January 1st, 2017.
- 48 Andreas Antonopoulos. *Mastering Bitcoin*. 2nd Edition, 2017.
- 49 Blockchain size. <https://blockchain.info/fr/charts/blocks-size>. Accessed on November 16th, 2017.
- 50 Number of orphaned blocks. <https://blockchain.info/charts/n-orphaned-blocks>, accessed on November 16th, 2017.
- 51 Blockchain size. <https://blockchain.info/charts/blocks-size>, accessed on November 16th, 2017.
- 52 Blockchain size. <https://blockchain.info/charts/blocks-size>, accessed on November 17th, 2017.
- [53] Block #0. Blockchain explorer. <https://blockexplorer.com/block/000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f>.
- 54 Block #495223 Information. <https://www.smartbit.com.au/block/00000000000000000004e3c9d483093f88760b3c4c7083308785f6c880f81ab31>
- 55 Bitcoin difficulty. <https://bitcoinwisdom.com/bitcoin/difficulty>, Accessed on November 17th, 2017.
- 56 ASIC machine. https://www.alibaba.com/product-detail/Fast-Delivery-Asic-Miner-Bitcoin-Mining_60678218176.html
- 57 ASIC machine. https://www.alibaba.com/product-detail/Fast-Delivery-Asic-Miner-Bitcoin-Mining_60678218176.html

-
- [58] Mining Pool. <https://blockchain.info/pools>, accessed on December 20th, 2017.
- 59 Andreas Antonopoulos. Mastering Bitcoin. 2nd Edition, 2017.
- 60 ISO/IEC 27001:2005. <https://www.iso.org/standard/42103.html>, Accessed on November 18th, 2017.
- 61 MEHARI, <https://clusif.fr/mehari/>, accessed on December 24th, 2017.
- 62 Risk assessment with OCTAVE, <https://pecb.com/whitepaper/risk-assessment-with-octave>, accessed on March 15th, 2018.
- 63 Cramm risk assessment tool, <http://goupegay.hatenablog.com/entry/2018/02/16/134818>, accessed on March 15th, 2018.
- 64 Filipe Macedo and Miguel Mira da Silva. Comparative study of information security risk assessment models. <https://fenix.tecnico.ulisboa.pt/downloadFile/395139415147/resumo.pdf>, Accessed on November 20th, 2017.
- 65 European Union Agency for Network and Information Security. https://www.enisa.europa.eu/topics/threat-risk-management/risk-management/current-risk/risk-management-inventory/rm-ra-methods/m_ebios.html.
- 66 International Organization for Standardization. ISO/IEC 13335-1:2004. <https://www.iso.org/standard/39066.html>
- 67 Visa processing, <https://www.visaeurope.com/enabling-payments/processing>, accessed on October 12th, 2018.
- 68 Bitcoin scalability problem, https://en.wikipedia.org/wiki/Bitcoin_scalability_problem, accessed on October 12th, 2018.
- 69 Top 10 Cryptocurrencies With Fast Transaction Speeds, <https://coinsutra.com/transaction-speeds/>, accessed on October 12th, 2018.
- 70 Melanie Swan. 2015. Blockchain: Blueprint for a new economy. "O'Reilly Media, Inc."
- 71 Karame, G., Androulaki, E., Capkun, S.: Two Bitcoins at the Price of One? Double-Spending Attacks on Fast Payments in Bitcoin. In: Proc. of Conference on Computer and Communication Security. (2012)
- 72 Karame, G., Androulaki, E., Capkun, S.: Two Bitcoins at the Price of One? Double-Spending Attacks on Fast Payments in Bitcoin. In: Proc. of Conference on Computer and Communication Security. (2012)
- 73 Github.com....
- 74 BIP 141, <https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki>, accessed on October 12th, 2018.
- 75 Beikverdi A, Song J. Trend of centralization in Bitcoin's distributed network, 2015.
- 76 Ghassan KARAME & Elli ANDROULAKI, Bitcoin and Blockchain Security, 2016.
- 77 Ittay Eyal, Emin Gün Sirer: Majority is not enough: bitcoin mining is vulnerable. Commun. ACM 61(7): 95-102 (2018)
- 78 GHASSAN O. KARAME & ELLI ANDROULAKI, Misbehavior in Bitcoin: A Study of Double-Spending and Accountability
- 79 Ghassan KARAME & Elli ANDROULAKI, Bitcoin and Blockchain Security, 2016.
- 80 Ghassan KARAME & Elli ANDROULAKI, Bitcoin and Blockchain Security, 2016
- 81 Vitalik Buterin, Ethereum White Paper A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM, https://cryptorating.eu/whitepapers/Ethereum/Ethereum_white_paper.pdf, accessed on October 14th, 2018
- 82 Yes, Bitcoin Can Do Smart Contracts and Particl Demonstrates How, <https://bitcoinmagazine.com/articles/yes-bitcoin-can-do-smart-contracts-and-particl-demonstrates-how/>, accessed on October 14th, 2018
- 83 Bitcoin Difficulty. <https://en.bitcoin.it/wiki/Difficulty>. Accessed on December 26th, 2017
- 84 Block Explorer, <https://blockexplorer.com/block/00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048>. Accessed on December 27th, 2017
- 85 Andreas M. Antonopoulos. Mastering Bitcoin: Programming the Open Blockchain 2nd Edition, 2017.
- 86 Block Explorer, <https://blockexplorer.com/block/000000000000000000000004e3c9d483093f88760b3c4c7083308785f6c880f81ab31>. Accessed on August 10th, 2018.
- 87 Bitcoin difficulty. <https://bitcoinwisdom.com/bitcoin/difficulty>. Accessed on December 27th, 2017.
- 88 Bitcoin 51% attack, <https://Bitcoin.org/en>, accessed on October 11th, 2018
- 89 G.O Karame, "Two bitcoins at the price of one? Double-spending attacks on fast payments in bitcoin", 2016
- 90 M. Rosenfeld, "Analysis of hashrate-based double spending", 2014
- 91 I.C Lin and T.C Liao, "A survey of Blockchain Security Issues and Challenges", 2017
- 92 Andreas. Antonopoulos, 51% Bitcoin attack, <https://www.youtube.com/watch?v=ncPyMUfNyVM>
- 93 Dining philosophers problem, <https://www.studytonight.com/operating-system/dining-philosophers-problem>
- 94 Deepshikha Bhargava ; Sonali Vyas Agent based solution for dining philosophers problem
- 95 Philosophers at a table, <http://qstuff.blogspot.com/2007/01/philosophers-at-table.html>, accessed on 28th May 2018.
- 96 Dining philosophers problem, <http://www.cs.fsu.edu/~baker/opsys/notes/philos.html>
- 97 Dining Philosophers problem deadlock, https://www.youtube.com/watch?v=_ruovgwXyYs
- 98 M. Chandy and J. MISRA, The drinking Philosophers Problem, 1984.
- 99 Dining philosophers problem Dijkstra's solution, <https://gist.github.com/glts/0bb56d89e7d4597cd4ec>
- 100 Prism model checker, <http://www.prismmodelchecker.org/tutorial/phil.php>
- 101 Prism, <http://www.prismmodelchecker.org/>
- 102 Adaptation of the dining philosophers problem, <http://www.prismmodelchecker.org/casestudies/phil.php>
- 103 Dining philosophers' problem solution, <http://www.baeldung.com/java-dining-philosophers>
- 104 Standards for efficient Cryptography, <http://www.secg.org/sec2-v2.pdf>
- 105 Bitcoin Wallet Encryption, https://en.bitcoin.it/wiki/Wallet_encryption
- 106 Andreas M. Antonopoulos, Mastering Bitcoin, 2nd edition, 2017.
- 107 Bitcoin wiki: Elliptic Curve Digital Signature Algorithm. https://en.bitcoin.it/wiki/Elliptic_Curve_Digital_Signature_Algorithm
- 108 Ghassan Karame , Elli Audroulaki, Bitcoin and Blockchain Security, 2016
- 109 US Army Joint Publication, http://www.jcs.mil/Portals/36/Documents/Doctrine/pubs/jp5_0_20171606.pdf
- 110 Base58 symbol chart used in Bitcoin. https://en.bitcoin.it/wiki/Base58Check_encoding. Accessed on June 8th, 2017.
- 111 Coin market capitalization. <https://coinmarketcap.com/currencies/bitcoin/#charts>. Accessed on December 26th, 2017
- 112 Certicom Research. Standards for Efficient Cryptography. SEC 2: Recommended Elliptic Curve Domain Parameters. (n.d.). Retrieved from <http://www.secg.org/sec2-v2.pdf>. Accessed on December 26th, 2017
- 113 Raja Sakti Arief Daulay et al 2017 IOP Conf. Ser.: Mater. Sci. Eng. 260 012002

B. Lamiri, K. Gueraoui, G. Zeggwagh

- 114 Garay J., Kiayias A., Leonardos N. (2017) The Bitcoin Backbone Protocol with Chains of Variable Difficulty. In: Katz J., Shacham H. (eds) Advances in Cryptology – CRYPTO 2017. CRYPTO 2017. Lecture Notes in Computer Science, vol 10401. Springer, Cham.
- 115 Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert van Renesse. , Cornell University. Bitcoin-NG: A Scalable Blockchain Protocol. <https://www.usenix.org/system/files/conference/nsdi16/nsdi16-paper-eyal.pdf>. Accessed on August 10th, 2018.
- 116 Bitcoin Difficulty. <https://en.bitcoin.it/wiki/Difficulty>. Accessed on December 26th, 2017
- 117 Block Explorer, <https://blockexplorer.com/block/00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048>. Accessed on December 27th, 2017
- 118 Andreas M. Antonopoulos. Mastering Bitcoin: Programming the Open Blockchain 2nd Edition, 2017.
- 119 Block Explorer, <https://blockexplorer.com/block/00000000000000000004e3c9d483093f88760b3c4c7083308785f6c880f81ab31>. Accessed on August 10th, 2018.
- 120 Bitcoin difficulty. <https://bitcoinwisdom.com/bitcoin/difficulty>, accessed on December 27th, 2017.
- 121 Coin market capitalization. <https://coinmarketcap.com/currencies/bitcoin/#charts>. Accessed on December 26th, 2017.
- 122 Certicom Research. *Standards for Efficient Cryptography. SEC 2: Recommended Elliptic Curve Domain Parameters*. (n.d.). Retrieved from <http://www.secg.org/sec2-v2.pdf>.
- 123 Brito, J and Van Valkenburgh, P. Bitcoin: Risk Factors for Insurance. <https://www.lloyds.com/news-and-insight/risk-insight/library/technology/bitcoin>
- 124 Bitcoin 51% attack, <https://Bitcoin.org/en>, accessed on June 6th, 2018
- 125 Ghassan Karame, Elli Androulaki, Srdjan Capkun: Two Bitcoins at the Price of One? Double-Spending Attacks on Fast Payments in Bitcoin. IACR Cryptology ePrint Archive 2012: 248 (2012)
- 126 Meni Rosenfeld: Analysis of Hashrate-Based Double Spending. CoRR abs/1402.2009 (2014)
- 127 Iuon-Chang Lin, Tzu-Chun Liao: A Survey of Blockchain Security Issues and Challenges. I. J. Network Security 19(5): 653-659 (2017).
- 128 Martijn Bastiaan: Preventing the 51%-Attack: a Stochastic Analysis of Two Phase Proof of Work in Bitcoin (2015). <https://pdfs.semanticscholar.org/0336/6d1fda3b24651c71ec6ce21bb88f34872e40.pdf>, accessed on January 4th, 2018.
- 129 Ittay Eyal, Emin Gün Sirer: Majority is not enough: bitcoin mining is vulnerable. Commun. ACM 61(7): 95-102 (2018).
- 130 Arthur Gervais, Hubert Ritzdorf, Ghassan Karame, Srdjan Capkun: Tampering with the Delivery of Blocks and Transactions in Bitcoin. ACM Conference on Computer and Communications Security 2015: 692-705.
- 131 Ayelet Sapirshstein, Yonatan Sompolsky, Aviv Zohar: Optimal Selfish Mining Strategies in Bitcoin. CoRR abs/1507.06183 (2015).
- 132 Zhenzhen Jiao, Rui Tian, Dezhong Shang and Hui Ding. Bicom: A Bilayer Scalable Nakamoto Consensus Protocol, CoRR, 2018, <https://arxiv.org/abs/1809.01593>.
- 133 Rui Tian, Wei Gong: Resisting Selfish Mining Attacks in the Bicom. CoRR abs/1809.06289 (2018).
- 134 Jaewon Bae, Hyuk Lim: Random Mining Group Selection to Prevent 51% Attacks on Bitcoin. DSN Workshops 2018: 81-82
- 135 Bitcoin Difficulty. <https://en.bitcoin.it/wiki/Difficulty>
- 136 Adapted from blockchain.info
- 137 June Ma Joshua S. Gans Rabee Tourky, MARKET STRUCTURE IN BITCOIN MINING, 2018
- 138 Hashrate Distribution, <https://blockchain.info/pools>
- 139 J.Bonneau Andrew Miller Jeremy Clark Arvind Narayanan Joshua A. Kroll Edward W. Felten, Research Perspectives and Challenges for Bitcoin and Cryptocurrencies
- 140 Dining philosophers problem, <https://www.studytonight.com/operating-system/dining-philosophers-problem>
- 141 Deepshikha Bhargava ; Sonali Vyas
Agent based solution for dining philosophers problem.
- 142 Philosophers at a table, <http://qstuff.blogspot.com/2007/01/philosophers-at-table.html>, accessed on 28th May 2018.
- 143 Dining philosophers problem, <http://www.cs.fsu.edu/~baker/opsys/notes/philos.html>
- 144 Dining Philosophers problem deadlock, https://www.youtube.com/watch?v=_ruovgwXyYs
- 145 M. Chandy and J. MISRA, The drinking Philosophers Problem, 1984.
- 146 Dining philosophers problem Dijkstra's solution, <https://gist.github.com/glts/0bb56d89e7d4597cd4ec>
- 147 Prism model checker, <http://www.prismmodelchecker.org/tutorial/phil.php>
- 148 jPrism, <http://www.prismmodelchecker.org/>
- 149 Adaptation of the dining philosophers problem, <http://www.prismmodelchecker.org/casestudies/phil.php>
- 150 Dining philosophers' problem solution, <http://www.baeldung.com/java-dining-philosophers>
- 151 S. Barber, X. Boyen, E. Shi and E. Uzun. Bitter to Better — How to Make Bitcoin a Better Currency. Financial Cryptography 2012.
- 152 J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll and E. W. Felten, "SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies," 2015 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2015, pp. 104-121.
- 153 Mauro Conti, E Sandeep Kumar, Chhagan Lal, Sushmita Ruj. A Survey on Security and Privacy Issues of Bitcoin. ArXiv, 2017.
- 154 Dorit Ron and Adi Shamir. Quantitative analysis of the full bitcoin trans-action graph. In Proceedings of Financial Crypto 2013, 2013
- 155 E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized anonymous payments from Bitcoin. In IEEE Symposium on Security and Privacy, 2014
- 156 I, Miers, C. Garman, M. Green, and A. D. Rubin. Zerocoin: Anonymous Distributed E-Cash from Bitcoin. In IEEE Symposium on Security and Privacy, 2013.
- 157 Y Sompolsky, A Zohar. Bitcoin's security model revisited. ArXiv preprint arXiv:1605.09193
- 158 Gavin Wood. Ethereum: A secure decentralized generalised transaction ledger. Ethereum Project. Yellow Paper, 151, 2014. See also : <https://www.ethereum.org/>
- 159 J. Garay, A. Kiayias, and N. Leonardos. The Bitcoin Backbone Protocol: Analysis and Applications. Cryptology ePrint Archive, Report 2014/765, 2014.

- 160 Gramoli, V. (2018). From blockchain consensus back to Byzantine consensus [Forthcoming]. Future Generation Computer Systems, In Press.
- 161 Standards for efficient Cryptography, <http://www.secg.org/sec2-v2.pdf>
- 162 Bitcoin Wallet Encryption, https://en.bitcoin.it/wiki/Wallet_encryption
- 163 Andreas M. Antonopoulos, Mastering Bitcoin, 2nd edition, 2017.
- 164 Bitcoin wiki: Elliptic Curve Digital Signature Algorithm. https://en.bitcoin.it/wiki/Elliptic_Curve_Digital_Signature_Algorithm
- 165 Ghassan Karame , Elli Audroulaki, Bitcoin and Blockchain Security, 2016.
- 166 US Army Joint Publication, http://www.jcs.mil/Portals/36/Documents/Doctrine/pubs/jp5_0_20171606.pdf