Université de Limoges

Projet d'Initiation la Recherche

Master 1 Cryptis

La cryptographie appliquées à la cryptomonnaie

Réalisé par:
Hajar Boudfor

Encadrant:
Simone Naldi

Janvier,2023

Table des matières

1 Introduction

Avec les méthodes traditionnelles de paiement, nous sommes obligés de passer par un tiers de confiance (banque, etc.) pour réaliser une transaction. Cela présente plusieurs inconvénients : nous devons payer des frais pour ce service, et nous n'avons pas la garantie que le tiers de confiance ne disparaîtra pas avec notre argent.

En 2008, Satoshi Nakamoto a développé une solution de paiement électronique entièrement décentralisée, appelée bitcoin, qui permet de faire des transactions sans passer par un tiers de confiance. Le bitcoin a été lancé en janvier 2009 et permet de transférer de l'argent directement d'une personne à une autre, sans frais supplémentaires et avec une sécurité accrue.

2 Introduction à la cryptomonnaie et à la cryptographie

2.1 Définition de la cryptomonnaie et de son fonctionnement

Une cryptomonnaie est une monnaie numérique qui utilise la technologie de la chaîne de blocs (blockchain) pour sécuriser les transactions et empêcher la double dépense. Elle repose sur l'utilisation de la cryptographie pour protéger les échanges et créer de la valeur. Les cryptomonnaies sont décentralisées, c'est-à-dire qu'elles ne sont pas contrôlées par une banque centrale ou tout autre autorité. Elles peuvent être utilisées pour effectuer des paiements en ligne et pour stocker de la valeur comme n'importe quelle monnaie traditionnelle. Il existe de nombreuses cryptomonnaies différentes, chacune ayant ses propres caractéristiques et utilisations. Bitcoin est la plus connue et la plus ancienne, mais il y a aussi Ethereum, Litecoin, Monero, et de nombreuses autres.

La plupart des cryptomonnaies fonctionnent grâce à la technologie de la blockchain, qui est une base de données distribuée qui utilise des fonctions cryptographiques pour vérifier et enregistrer des informations. La blockchain a été initialement développée pour résoudre le problème des "généraux byzantins", qui consiste à trouver un moyen pour que des entités puissent communiquer des informations même si certaines d'entre elles sont défaillantes ou malveillantes. Dans le protocole Bitcoin, la blockchain est utilisée pour éviter les actes malveillants tels que la double dépense, en enregistrant toutes les transactions dans un registre partagé par tous les nœuds du réseau. Pour inscrire un nouveau bloc dans la blockchain, une preuve de travail (ou "proof of work") est requise, qui consiste en des calculs complexes nécessitant des ressources conséquentes. Les mineurs, qui sont chargés de réaliser une partie de cette preuve de travail, sont récompensés en Bitcoins pour leur contribution à la sécurisation du réseau.

2.2 Importance de la sécurité dans les transactions de cryptomonnaie

La technologie de blockchain est utilisée pour sécuriser les transactions de cryptomonnaie. C'est est une structure de données qui a des propriétés de sécurité intégrées grâce La technologie de blockchain est utilisée pour sécuriser les transactions de cryptomonnaie. C'est est une structure de données qui a des propriétés de sécurité intégrées grâce à sa base en cryptographie, décentralisation et consensus. Elle est utilisée pour enregistrer des transactions ou d'autres

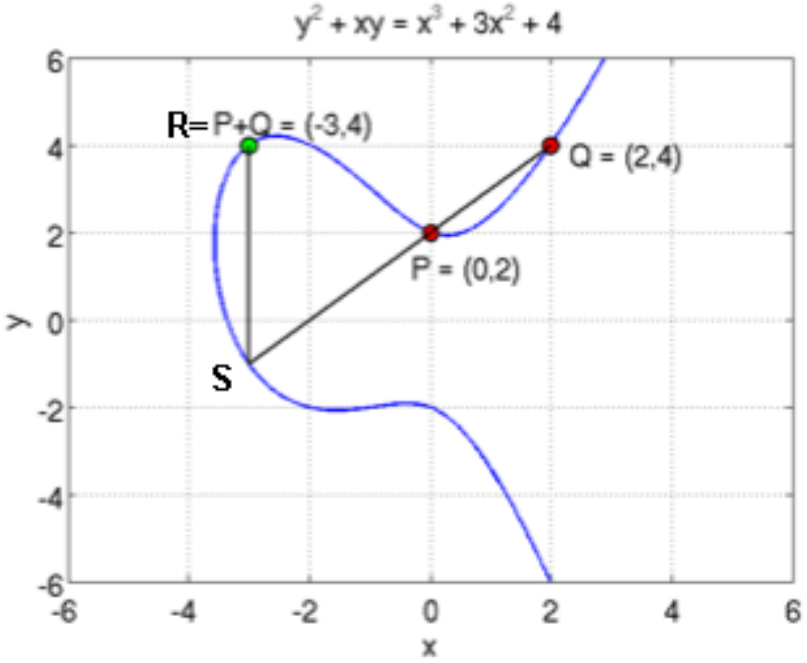
données dans des blocs qui sont liés par une chaîne cryptographique, ce qui rend presque impossible toute modification des données.

Toutes les transactions doivent être validées et approuvées par un mécanisme de consensus pour garantir leur véracité et exactitude.

La technologie de blockchain permet la décentralisation grâce à la participation de différents membres dans un réseau distribué, sans point de défaillance unique et sans possibilité pour un utilisateur unique de modifier l'enregistrement des transactions. Cependant, les technologies de blockchain peuvent varier en termes de sécurité sur certains aspects.

2.3 Présentation de l'algorithme ECDSA et de son rôle dans la sécurisation des transactions de bitcoin

ECDSA signifie bien Algorithme de signature numérique à courbe elliptique. C'est un algorithme de cryptographie utilisé pour générer et vérifier des signatures numériques, c'est-à-dire des codes qui permettent de vérifier l'intégrité d'une donnée numérique et d'authentifier l'identité de son auteur. ECDSA utilise la cryptographie à courbe elliptique pour accomplir cette tâche, ce qui lui permet de générer et de vérifier des signatures de manière plus sécurisée et efficace que d'autres algorithmes de signature numérique.



En bitcoin, l'algorithme de signature numérique à courbe elliptique (ECDSA) joue un rôle important dans la sécurisation des transactions. Lorsqu'un utilisateur envoie des bitcoins à un autre utilisateur, il doit signer numériquement la transaction avec sa clé privée pour prouver qu'il est bien le propriétaire des bitcoins qu'il envoie. La signature numérique générée avec ECDSA

est vérifiée par les autres nœuds du réseau bitcoin pour s'assurer que la transaction est valide et que les bitcoins ne sont pas envoyés de manière frauduleuse. ECDSA est également utilisé pour générer et vérifier les clés publiques et privées qui sont utilisées pour envoyer et recevoir des bitcoins. En résumé, ECDSA joue un rôle crucial dans la sécurisation des transactions bitcoin et dans la vérification de l'identité des utilisateurs du réseau.

3 Fonctionnement de l'ECDSA

3.1 Explication du fonctionnement de l'ECDSA en utilisant des clés publiques et privées

L'ECDSA (Elliptic Curve Digital Signature Algorithm) est un algorithme de cryptographie utilisé pour signer et vérifier des données de manière sécurisée. Il repose sur l'utilisation de trois éléments clés : la clé privée, la clé publique et la signature.

La clé privée est un nombre secret généré aléatoirement qui est utilisé pour signer des données.Elle est connue uniquement de la personne qui l'a créée et est utilisée pour dépenser les fonds dans la blockchain de Bitcoin.

La clé publique est un nombre généré à partir de la clé privée grâce à une relation mathématique complexe. Elle est utilisée pour vérifier la signature et doit être partagée publiquement.

La signature est un nombre généré mathématiquement à partir du hachage des données à signer et de la clé privée. Elle est constituée de deux parties, "r" et "s", et peut être vérifiée en utilisant la clé publique et un algorithme mathématique.

Grâce à ces trois éléments, l'ECDSA garantit l'authenticité des données signées et empêche leur modification non autorisée.

3.2 Exemple de chiffrement et déchiffrement d'une transaction de Bitcoin

1. Jacqueline veut envoyer 0,1 bitcoin à Robert. Elle commence par créer une demande de transaction qui spécifie l'adresse de destination (c'est-à-dire celle de Robert), le montant de la transaction (0,1 bitcoin) et l'adresse de Jacqueline (qui servira de preuve de sa propriété du bitcoin qu'elle envoie).

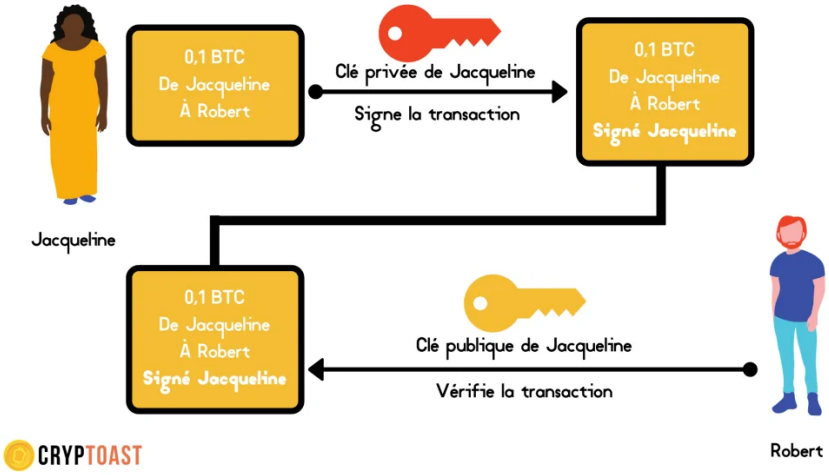
2. Jacqueline utilise sa clé privée pour signer la demande de transaction. La signature est générée en utilisant l'algorithme ECDSA et permet de prouver que Jacqueline est bien le propriétaire du bitcoin qu'elle envoie.

3. Jacqueline envoie la demande de transaction signée à la blockchain.

4. La blockchain vérifie la validité de la transaction en utilisant l'algorithme ECDSA. Pour ce faire, elle utilise la clé publique de Jacqueline qui est associée à sa clé privée utilisée pour signer la transaction. Si la signature est valide, cela prouve que Jacqueline est bien le propriétaire du bitcoin qu'elle envoie et que la transaction est autorisée.

5. Si la transaction est valide, elle est ajoutée à la blockchain et elle devient publique.

6. Robert peut maintenant utiliser sa clé privée pour déchiffrer la transaction et vérifier que c'est bien 0,1 bitcoin qui lui a été envoyé par Jacqueline.

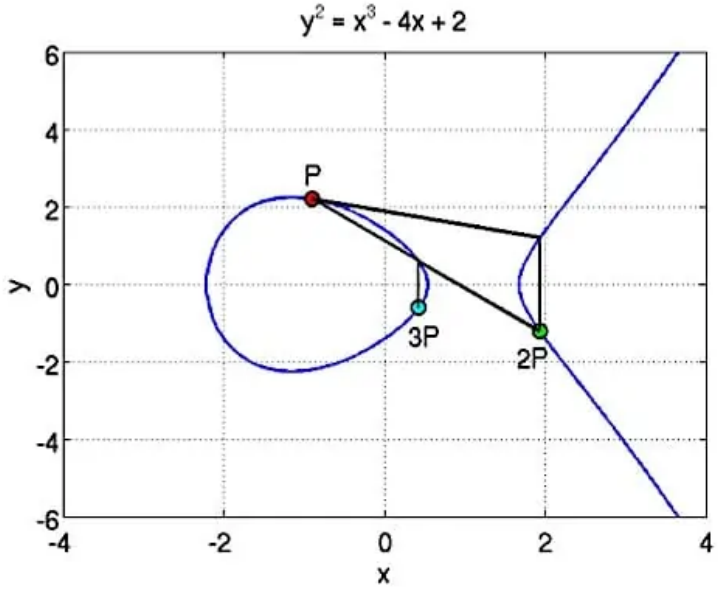


3.3 Processus de génération et de gestion des clefs dans l'ECDSA

Dans l'algorithme ECDSA, les clefs (clé publique et clé privée) sont générées et gérées de la manière suivante :

- **Génération de la clef** : pour créer une paire de clefs, on utilise un processus appelé "génération de clé". Cela implique de choisir aléatoirement deux nombres : la clef privée et un point sur l'ellipse de courbe elliptique sélectionnée. Ce point est appelé "clef publique".
- **Signature de la clef** : pour signer une donnée (par exemple une transaction Bitcoin), on utilise la clef privée. L'algorithme ECDSA est utilisé pour générer une signature qui prouve l'authenticité de la donnée et vérifie qu'elle n'a pas été modifiée.
- **Vérification de la clef** : pour vérifier la signature d'une donnée, on utilise la clef publique. L'algorithme ECDSA permet de vérifier que la signature est valide et que la donnée n'a pas été modifiée.

Voici un exemple de génération de clefs:



Sur cette courbe, un point est choisi au hasard et est considéré comme son point d'origine. Ensuite, un nombre aléatoire est généré, c'est ce nombre aléatoire, qui représentera la clé privée. Ensuite, en utilisant la clé privée et le point d'origine, on effectue une autre équation et on obtient un deuxième point sur la courbe, c'est la clé publique. L'utilisation de cette nouvelle équation avec le point d'origine et la clé publique nous permet ainsi d'établir la relation entre les clés publique et privée.

Ce processus est considéré comme sûr, car pour le moment, il ne peut être effectué que dans une seule direction

De cette façon, lorsqu'un utilisateur veut signer un fichier, il utilise sa clé privée, qui est un nombre aléatoire, avec un hachage du fichier (un numéro unique qui représente le fichier) pour créer une signature. Si quelqu'un souhaite vérifier l'authenticité de cette signature, il n'a besoin que de la clé publique de l'utilisateur, qui peut être connue de tous sans compromettre la sécurité de la signature, car la clé publique ne sert qu'à vérifier, pas à signer.

4 Sécurité de l'ECDSA

4.1 Intégration et sécurité

En pratique, l'ECDSA est souvent basée sur des courbes recommandées par des organisations telles que NIST et Certicom.

Le NIST recommande par exemple quinze courbes elliptiques différentes sur dix corps différents. Cinq courbes sont recommandées sur cinq corps finis d'ordre p premier$\mathbb{F}_p$, nommées P-192, P-224, P-256, P-384, P-521, dix courbes sur cinq corps finis de la forme $\mathbb{F}_{2^m}$

L'ANSSI recommande l'utilisation de la courbe FRP256v1, dont les paramètres ont été publiés

au Journal Officiel7 en 2011, et les courbes P-256, P-384, P-521, B-283, B-409 et B-571 définies dans le FIPS 186-2.

Puisque tous les algorithmes connus pour résoudre le problème du logarithme discret sur les courbes elliptiques sont en $O(\sqrt{n})$ (Baby-step giant-step, L'algorithme de rho Pollard), la taille du corps doit donc être approximativement deux fois plus grande que le paramètre de sécurité voulu. Pour un degré de sécurité de 128-bits (AES-128, RSA-3072), on prendra une courbe sur un corps $\mathbb{F}_q$, où $q \approx 2^{256}$.

4.2 Analyse de force de l'ECDSA en termes de sécurité

l'utilisation de l'authentification cryptographique à courbe elliptique offre une meilleure efficacité par rapport à RSA parce que les clés de plus courte longueur peut être utilisé sans que la sécurité du système soit compromise. Dans l'ECDSA, la taille des bits de la clé publique est deux fois la taille du paramètre de sécurité, en bits.

L'attractivité de l'ECDSA réside dans le fait qu'il n'y a pas algorithme sous-exponentiel connu pour résoudre le ECDLP sur un correctement choisi courbe elliptique. Ainsi, il prend plein exponentiel temps pour résoudre le PPEL par rapport au RSA où le algorithmes les plus connus pour résoudre le nombre entier sous-jacent problème de factorisation prend un temps sous-exponentiel. Cela signifie que des paramètres importants plus petits avec une sécurité équivalente peut être utilisé dans ECDSA que dans RSA. Certains avantages sont plus rapides calculs, réduction de la puissance de traitement, réduction du stockage espace et bande passante. Cela rend ECDSA très idéal pour TTP. Ci-dessous, nous donnons un tableau, en comparant la taille de la clé ECDSA avec l'équivalent RSA. Le tableau ci-dessous montre la taille de la clé comparaison pour ECDSA et RSA

ECDSA	160	224	256	384	512
RSA	1024	2048	3072	7680	15360

4.3 Attaques contre l'ECDSA

Les attaques possibles sur ECDSA peut être classé comme suit :

4.3.1 Attaques sur le problème de logarithme discret de la courbe elliptique

. L'attaque de signature à médian *(middle-signature attack)* vise à créer une signature valide pour un message choisi en utilisant la signature d'un autre message. Cette attaque exploite une propriété de la courbe elliptique qui permet de calculer facilement un point intermédiaire entre deux points connus sur la courbe.

Pour réaliser cette attaque, l'attaquant doit connaître la signature d'un message quelconque et choisir un autre message pour lequel il souhaite créer une signature valide. En utilisant la propriété de la courbe elliptique mentionnée précédemment, l'attaquant peut alors calculer un point intermédiaire entre les deux points représentant les signatures du premier et du second

message. En utilisant ce point intermédiaire, l'attaquant peut créer une signature valide pour le second message.

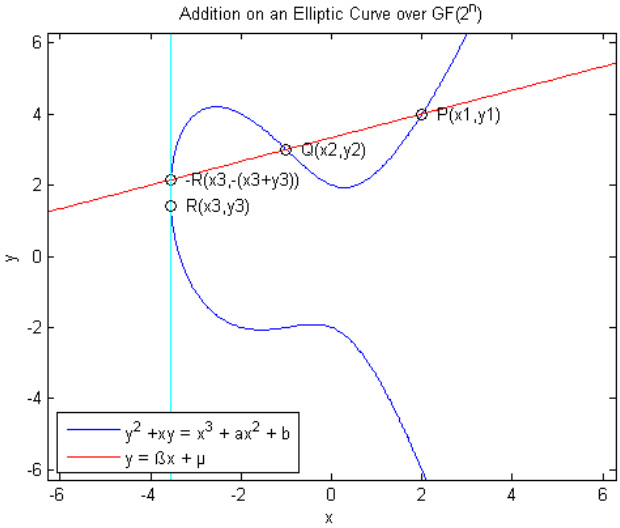
Cette attaque peut être utilisée pour créer de fausses signatures valides pour des messages choisis, ce qui peut compromettre la sécurité de l'ECDSA. Pour se protéger contre cette attaque, il est important de choisir des paramètres de sécurité appropriés et de vérifier régulièrement l'intégrité des signatures. Il est également recommandé de mettre en place des contrôles pour détecter les signatures suspectes ou les messages associés à de fausses signatures.

4.3.2 Attaques sur la fonction de hachage utilisée

Le « paradoxe des anniversaires » est le fait que, dans un groupe d'au moins 23 personnes, la probabilité qu'il y ait deux personnes dont les anniversaires tombent le même jour est supérieure à 1/2 (ce qui est assez surprenant à première vue, d'où le terme de 'paradoxe', à ne pas prendre au sens de la logique bien sûr). En considérant la fonction h qui à une personne associe sa date d'anniversaire, trouver deux personnes dont l'anniversaire tombe le même jour revient à trouver une collision pour h.
Pour une fonction de hachage h dont l'empreinte est de taille l (donc avec 2 empreintes possibles), on montre que la probabilité de trouver une collision en calculant les images par h d'environ $2^{\frac{l}{2}}$ éléments de l'ensemble de départ est au moins 1/2. On obtient donc une attaque, dite du paradoxe des anniversaires, consistant à calculer les empreintes de $2^{\frac{l}{2}}$ messages tirés au hasard, avec au moins 1 chance sur 2 de trouver une collision. Assurer la résistance forte aux collisions, donc à cette attaque, oblige à choisir ' suffisamment grand (160 bits dans le cas de SHA-1, au moins 256 bits aujourd'hui).

4.4 Évaluation du protocole de sécurité dans le contexte de la cryptomonnaie Bitcoin

- La courbe utilisée par bitcoin est une courbe dite **secp256k1** ou Koblitz du nom du cryptographe, Neal Koblitz, qui, en 1985, a démontré l'utilité des courbes elliptiques en cryptographie.



secp256k1 n'a presque jamais été utilisé avant Bitcoin est devenu populaire, mais il gagne maintenant en popularité en raison de ses plusieurs propriétés agréables. Les courbes les plus couramment utilisées ont une structure aléatoire, mais secp256k1 a été construit d'une manière spéciale non aléatoire qui permet un calcul particulièrement efficace. Par conséquent, elle est souvent plus rapide de 30 % que les autres courbes si l'implémentation est suffisamment optimisée. En outre, contrairement aux courbes NIST populaires, les constantes de secp256k1 ont été sélectionnées de manière prévisible, ce qui réduit considérablement la possibilité que le créateur de la courbe ait inséré une sorte de porte dérobée dans la courbe.

- Le protocole Bitcoin repose sur un algorithme de chiffrement appelé SHA-256, soit Secure Hash Algorithm (algorithme de hachage sécurisé). Les ordinateurs et algorithmes actuels, dont la puissance de calcul est limitée, sont incapables de contourner cet algorithme de chiffrement par une attaque par force brute. Dans ce type d'attaque, l'attaquant utilise un logiciel pour tester des milliards de combinaisons jusqu'à tomber sur le bon mot de passe, ou la bonne clé privée, permettant d'accéder à un portefeuille numérique. Depuis sa création en 2009 par le mystérieux Satoshi Nakamoto, le Bitcoin s'est donc avéré infaillible.

5 Autres applications de l'ECDSA

- Systèmes de gestion de droits d'accès : l'ECDSA est utilisé pour vérifier l'identité de l'utilisateur et contrôler l'accès aux ressources et aux systèmes protégés par mot de passe.
- Applications mobiles : l'ECDSA est utilisé pour sécuriser les communications entre les applications mobiles et les serveurs, ainsi que pour l'authentification de l'utilisateur.
- Internet des objets (IoT) : l'ECDSA est utilisé pour sécuriser les communications entre les appareils de l'IoT et les serveurs, ainsi que pour l'authentification de l'utilisateur.

- Gestion de la confidentialité : l'ECDSA est utilisé pour protéger la confidentialité des données sensibles, telles que les données médicales ou financières, en utilisant des signatures numériques pour vérifier l'intégrité des données et s'assurer qu'elles n'ont pas été altérées.

6 conclusion

L'algorithme ECDSA basé sur les courbes elliptiques se posent en alternative efficace face à l'incontournable RSA. En effet, il exploite un problème mathématique différent, qui est réputé pour sa solidité égale à RSA pour des clés de longueur bien inférieure. Cela les rend parfaitement adaptés à la signature numérique des données envoyées au cours d'une communication, afin de vérifier l'intégrité de ces données et de s'assurer qu'elles n'ont pas été altérées pendant le transport.

Pour conclure, notre recherche nous a permis d'acquérir une bonne compréhension de l'algorithme ECDSA et de son rôle dans la sécurisation des transactions de Bitcoin. Afin de poursuivre l'exploration de ce sujet, nous souhaitons étudier la vulnérabilité de l'ECDSA au cours du second semestre. Nous visons à identifier et à examiner les différentes attaques connues contre l'ECDSA pour évaluer les risques potentiels pour les utilisateurs de crypto-monnaies qui utilisent cet algorithme de chiffrement.

References

[1] https://www.xmco.fr/actu-secu/XMCO-ActuSecu-49-Dossier_Cryptomonnaies.pdf

[2] Johnson, D., Menezes, A. Vanstone, S (2001). The Elliptic Curve Digital Signature Algorithm (ECDSA) *Springer-Verlag 2001*

[3] Ballet, S. Bonecaze, A. (2011-2012). Cours de cryptographie avancée *Courbes elliptiques: application à la cryptographie*

[4] Douglas, Stinson. (2002). Cryptography-Theory and Practice. CRC Press.

[5] https://www.lama.univ-savoie.fr/mediawiki/images/f/f2/Carbonnier-Arrigo.pdf

https://e-ducat.fr/links/ecdsa/

[7] https://academy.bit2me.com/fr/que-es-ecdsa-curva-eliptica/les-bases-de-l'ecdsa

[8] https://unglueit-files.s3.amazonaws.com/ebf/05db7df4f31840f0a873d6ea14dcc28d.pdf

[9] https://www.ibm.com/fr-fr/topics/blockchain-security

[10] Vinatier, S (2022). Cours d'introduction à la cryptographie *M1 Cryptis*