The Advance of Cryptocurrency Wallet with Digital Signature

Pu Ji *

College of Software and Engineering, Miami University, Oxford, Ohio, USA

* Corresponding author email: jip@miamioh.edu

Abstract. This paper explains the composition of the cryptocurrency wallet by analyzing a wallet's components, functions, categories, the algorithm of signing, and security features as three major components used in the constitute of the cryptocurrency wallet, private key, public key, and address are introduced from their generation process with specific Elliptic Curve Digital Signature Algorithm secp256k1 and Keccak hash function. A variety of crypto wallets are divided into different categories by their purpose of use, the form of display, and Internet connectivity, and multiple crypto wallets are explained in the second part of this paper. The elliptic curve secp256k1 is selected in Ethereum for both calculating the public key information and performing signing transactions. This paper details secp256k1 and Keystore used behind the signing process and security of cryptocurrency wallet and outlines potential risks due to the development of computing power and transfer of validation system in Ethereum.

Keywords: Cryptocurrency Wallet; Elliptic Curve Digital Signature Algorithm; Secp256k1; Keystore; Ethereum.

1. Introduction

A cryptocurrency wallet, as the fundamental tool used by users who intend to interact with the blockchain network and blockchain platform, is an indispensable part of performing transactions of cryptocurrency. Practically, the traditional wallet stores the physical form of value of any kind of currency such as coin, banknote, and precious metal. Cryptocurrency wallet, in comparison to the traditional wallet, does not store cryptocurrencies and can be considered as access that allows users to communicate with blockchain to store or verify the data of each transaction.

A cryptocurrency wallet in Ethereum has three essential components as the private key, public key, and wallet address [1]. The private key, as a core of a crypto-wallet, is a 256 bits integer that is generated randomly and displayed as a 64-digit hexadecimal string. According to the Ethereum yellow paper, a private key of a 256 bits integer is randomly generated while creating a crypto wallet. From a security perspective, the private key should be safely kept and available to users only. The public key of the wallet is a 512 bits integer, displayed in 128-digit hexadecimal string, and calculated based on the randomly generated private key through the elliptic curve digital signature algorithm (ECDSA), and the specific curve secp256k1 is selected both in Bitcoin and Ethereum. It is used for the authentication of related digital signatures to determine whether the information communicated in the transaction belongs to the right owner or not. The wallet address is a 160 bits integer displayed in 40-digit hexadecimal string and is a result of the calculation of the Keccak-256 hash function of the corresponding public key.

The Elliptic Curve Digital Signature Algorithm and Keystore are two important features behind crypto wallets that provide protection and security for private key information. ECDSA is used not only to generate the public key but also to act in the signing process. The complexity of resolving hard problems like discrete logarithm problems, random selection, and the limitations of present computing ability form a barrier to reversing users' private keys. On the other hand, the Password Based Key Derivation Function applied in Keystore provides encryption on private keys with an external decryption key. Based on that, potential risk thread the security of algorithms and a change in the Ethereum working system are discussed in a later section.

2. The Overview of Cryptocurrency Wallets

Varieties of cryptocurrency wallets are divided into multiple categories. A cryptocurrency wallet can be classified as a cold wallet or a hot wallet due to its connectivity with the Internet. A cold wallet can be any physical hardware like a paper wallet, hard drive, and USB device containing the private key [2, 3]. Users can apply encryption on most of the cold wallets to ensure safer storage of key information. It is designed to prevent hacking from the Internet since the process of signing transactions is transferred to users' devices and has zero connection to the online server at that time. Based on that, even if the adversary injects malware into users' computers, they cannot access or steal users' private keys. A hot wallet is a crypto wallet connected to the Internet, such as a desktop wallet, web application wallet, mobile wallet, and others [2, 3]. Since the hot wallet is directly connected to the Internet, there exist possible vulnerabilities for adversaries to attack users' accounts, which may lead to a leak of private keys and loss of digital property, especially for web-based wallets. However, compared to a cold wallet, in exchange for security, it allows users to remain online accounts for better convenience and faster reaction to transactions.

2.1 Hot Wallet Examples

2.1.1 Desktop Wallet

Desktop wallet is an application downloaded and installed on users' desktop computers or laptops [2, 3]. Users can create a crypto wallet by using the application to generate key pairs and store them in local hardware. The application was often developed with a concise and user-friendly interface to ensure easier access for users to interact with transactions. It provides key pairs, connects to the Internet, and receive address for sending cryptocurrency while users need to perform transactions. From a security perspective, users can use a combination of usernames and passwords and apply multi-factor authentication for greater protection. Additionally, the desktop wallet usually provides a seed phrase and a recovery phrase, a list of randomly generated words to help users recover access to the desktop wallet. However, storing key information on local hardware is relatively vulnerable because it exposes vulnerabilities to potential malware or viruses in users' devices.

2.1.2 Web Wallet

Web wallet has less requirement to access since the user can access web wallet through any trusted internet browser on any possible device instead of downloading and installing specialized software [2]. Third-party companies behind these web application wallets store users' wallet information and achieve the ownership of users' cryptocurrency property, requiring a higher level of trust between users and third parties. On the other hand, an alternative plan was designed to resolve this type of problem. Users can be active in the multi-sig contract to form multi-sig wallet, which ensures users achieve part of the account's ownership. The multi-sig wallet requires multiple keys to generate a digital signature and sign a transaction. From a security perspective, web application remains online all the time and are more vulnerable to potential malware and hackers. Hence, a web wallet is the least secure wallet to use. In addition, some third parties require personal information for registration, such as contact number and email address, so less anonymity is contained in the web wallet for security purposes.

2.1.3 Mobile Wallet

Mobile Wallet: Mobile wallet requires users to download and install applications on their phones to send or receive cryptocurrency [2]. It brings a higher level of convenience and flexibility to users to perform transactions. From the security perspective, however, a mobile wallet offers low security and privacy since the mobile device is more vulnerable to malware or viruses and is more likely to be physically damaged, lost, or stolen. In addition, mobile devices contain necessary personal information like identity and geological information, which may lead to less anonymity for transactions if it is under malicious activities.

2.2 Cold Wallet Examples

2.2.1 Paper Wallet

A paper wallet can be any piece containing key information written on it. Only the person with physical access to the paper can own that crypto wallet [2, 3]. From a security point of view, due to the property of material can be physically damaged, destroyed, lost, or stolen, which will suspend users' access to their crypto wallets.

2.2.2 Hardware Wallets

A hardware wallet is one of the most secure methods to store your private key information. Most hardware wallets, such as USB drives, are completely disconnected from the Internet to avoid exposing vulnerabilities to potential malware or hackers [2, 3]. On the other hand, increasing security will not affect the flexibility of using hardware wallets to perform transactions. Users only need to connect a hardware wallet with any computers or internet-enabled devices to sign and authorize a transaction. Moreover, the private key will remain safely kept in the hardware even if the device is connected to the Internet. However, it can also be lost or stolen, and most hardware devices require a certain cost.

3. Elliptic Curve Digital Signature Algorithm

The Elliptic Curve Digital Signature Algorithm (ECDSA) plays an essential role not only in generating public key information but also in the digital signature signing process. It offers a variety of Digital Signature Algorithm (DSA) that follows Elliptic Curve Cryptography (ECC). It is an asymmetric encryption methodology that provides better performance and security than Rivest-Shamir-Adleman (RSA) encryption and allows each recipient from the network to apply message authentication to verify the correctness of the identity of the message sender.

Ethereum, as one of the largest cryptocurrency platforms, uses the same specific elliptic curve model as Bitcoin, which is defined as secp256k1 according to Standards for Efficient Cryptography (SEC) [4]. It indicates the domain parameters over field F_p with a Koblitz curve as $T = (p, a, b, G, n, h)$ in the curve equation $y^2 = x^3 + ax + b$, in which the field F_p, a, b, and other parameters are defined within Table 1.

Table 1. Secp256k1 parameters over field F_p [4]

Field *p*	F_p = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFC2F = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1
a	*a* = 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
b	*b* = 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000007
Generator point (base point) in compressed form:	*G* = 02 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798
Generator point (base point) in uncompressed form:	*G* = 04 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798 483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8
Prime order *n* of *G*:	n = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141
Cofactor *h*:	*h* = 01

During the creation process of a crypto wallet, a large number (256-bit) integer is randomly selected in the interval $[1, n-1]$ as the secret private key of the wallet [5, 7]. By putting a randomly selected private key into the curve to achieve a coordinate (x, y) as public key by $K_{public} = K_{private} * G$. The entire process of generating the public key is irreversible, or in other words, is a

time-consuming problem to practically reverse the private key from the public coordinate. During the signing process, the message being sent will be firstly encrypted in the hash function to get a hashed value H_m. An ephemeral key k is randomly selected from the interval $[1, n-1]$ to calculate a coordinate (x, y) which is defined as $(x, y) = k * G$. The value of x-coordinate is used to calculate the value of r in which $r = x \ mod \ n$. If the value of r is equivalent to zero, another new ephemeral key k' is selected to iterate the whole process before calculating the value of r. Otherwise, s is calculated as $s = k^{-1} * (H_m + r * K_{private}) \ mod \ n$. Unless the value of s is equivalent to zero, which also needs to select another new ephemeral key k' to iterate the entire process, and the output signature is the pair value of (r, s).

4. Keystore

The Keystore file system exists in the Ethereum wallet as a solution to improve security and usability and is designed to provide an encrypted version of the private key and more accessible access with a passphrase set. Ethereum users can create individual passphrases instead of impractically memorizing an irregular 64-digit hexadecimal string. Specifically, the Keystore file stores essential data (cipher, cipherparams, ciphertext, kdf, Kd params, mac, id, version) in a JSON format. As a result, the Keystore file becomes a compromise between the security property and usability property.

A decryption key is used within the symmetric algorithm in the Keystore file to decrypt the ciphertext of the private key. Users' passphrases will be combined with key derivation function parameters dkLen, n, r, p, and salt as input of scrypt password-based key derivation function (PBKDF). To avoid brute force attacks, the salt value is used, as an extensive sequence of additional characters, to form a combination of salt and password. Moreover, more elements are added to KDF parameters to ensure it is costly for adversaries to perform large-scale hardware attacks. According to Dr. Colin Percival, n - cpu cost parameter, r - memory cost parameter, p - parallelization parameter, dkLen- output length is added to improve the complexity of scrypt PBKDF to combat parallel attacks [8]. The scrypt should first perform hash message authentication code HMAC for authentication by comparing the m value from HMAC SHA-256, in which $m = H\ (Pxoropad)|H((Pxoripad)|Data))$. It creates an object HMAC (h(password), null) and brings this object into PBKDF2. The PBKDF2 performs a loop containing a Pseudo Random Function to produce random messages to be fed into MAC to receive a return value. Then a loop of the SMIX algorithm with p iteration is started with the return value, and lastly, another PBKDF2 is performed to achieve the final decryption key. To access the private key, the encrypted ciphertext, cipher parameters, and decryption key are used as inputs of the specific symmetric algorithm AES-128-ctr for decryption. In addition, the MAC value stored in the Keystore file stands for Message Authentication Code, a type of digital signature used for authentication purposes. After inputting the password, a decryption key will be formed as described before through scrypt PBKDF, and the concatenation of the decryption key and ciphertext will be put into the SHA3-256 hash function to achieve a hash value as a result. Continuously, the message authentication code is compared with the hash result to prove whether the user provides a correct password.

5. Discussion

From the security perspective, although major cryptocurrencies and platforms such as Bitcoin and Ethereum have used secp256k1 for the past decade, the elliptic curve digital signature algorithm is never verified as a completely secured methodology. The complexity of breaking the digital signature signing algorithm depends on multiple factors, such as the discrete logarithm problem (DLP), SHA-3 hash function, and the property of random generating k [9]. The discrete logarithm problem remains a hard problem in cryptography, and it is extremely difficult to reverse a result into a giving factor. In practice, it has been seen as a time-consuming problem, and no successful case exists to resolve

the problem. Theoretically, with the ability of calculation that is the approach to large enough, it is possible to solve the discrete logarithm problem within a limited time to destroy the complexity caused by DLP. In that specific case, attackers can compute the DLP directly and further obtain the private key. The SHA-3 Keccak hash algorithm is considered one of the next decade's most secure and efficient hash algorithms. It requires a time that is too large to reverse the function and is capable of collision resistance and avoiding length extension attack.

Moreover, the temporary key in the signing process is randomly selected and unpredictable. Overall, these protection foundations rely on the limitation of current computing power. Further technology, such as quantum computing, has been discussed and discovered in progress to achieve a magnificent computing power which may have a massive impact and destroy all these security defenses. In addition, a player with higher computing power is more likely to pass the proof-of-work system and be rewarded with tokens. Ethereum, as one of the largest platforms around the world, 2022, has transferred from proof-of-work to proof-of-stake, which might be a part of the solution to this situation [10]. Instead of electing a winner that depends on the computing power, every single user with stake of tokens can participate in staking which may decreases the criteria to enter the competition than under proof-of-work. However, proof-of-stake requires accounts with a certain number of tokens or more to be selected as validators, and candidates with higher investment have larger possibilities of getting massive rewards. As a result, validators from the election such as Ethereum itself own a magnificent advantage from a huge number of stakes, and the change of working system pushed Ethereum approximate to centralization.

6. Summary

The cryptocurrency wallet has played a significant role in the blockchain network as a ticket that allows every user to access and be a part of more possibilities through the blockchain network. Many crypto wallets exist worldwide to satisfy users' purposes, such as cold storage like hardware and mobile wallets, representing properties of higher security but less flexibility and easier access to transactions but less anonymity, respectively. Therefore, maintaining the crypto wallet's security has become an essential target so that developments of elliptic curve algorithms and Keystore systems are studied to reduce and avoid potential risks as much as possible. The Elliptic Curve Digital Signature Algorithm is the core of protecting irreversibility with many mathematical hard problems such as discrete logarithm problems and random selection. Theoretically, however, there is still a possibility of breaking through the algorithm and attacking the wallet's private key information. This potential risk will enhance by the achievement of quantum computing, which can ensure a huge amount of computing power to resolve hard problems and take advantage of validator selection.

References

[1] Wood, G. Ethereum: A Secure Decentralised Generalized Transaction Ledger Berlin Version. Ethereum project yellow paper, Oct, 2022, https://ethereum.github.io/yellowpaper/paper.pdf.

[2] Suratkar, S., Shirole, M., & Bhirud, S. (2020, September). Cryptocurrency wallet: A review. In 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP) (pp. 1-7). IEEE.

[3] Khan, A. G., Zahid, A. H., Hussain, M., et al. (2019, November). Security of cryptocurrency using hardware wallet and qr code. In 2019 International Conference on Innovative Computing (ICIC) (pp. 1-10). IEEE.

[4] Daniel R. L. Brown. Standards for Efficient Cryptography 2 (SEC 2). Certicom Research, Jan, 2010, https:// www.secg.org/sec2-v2.pdf.

[5] Johnson, D., Menezes, A., & Vanstone, S. (2001). The elliptic curve digital signature algorithm (ECDSA). International journal of information security, 1(1), 36-63.

[6] Bi, W., Jia, X., & Zheng, M. (2018). A secure multiple elliptic curves digital signature algorithm for blockchain. arXiv preprint arXiv:1808.02988.

[7] Pote, S., Sule, V., & Lande, B. K. (2019, April). Arithmetic of Koblitz Curve Secp256k1 Used in Bitcoin Cryptocurrency Based on One Variable Polynomial Division. In 2nd International Conference on Advances in Science & Technology (ICAST).

[8] C. Percival, S. Josefsson. The Scrypt Password-Based Key Derivation Function. Internet Engineering Task Force (IETF), RFC 7914, Aug, 2016.

[9] Vaudenay, S. (2003). The Security of DSA and ECDSA. In: Desmedt, Y.G. (eds) Public Key Cryptography -- PKC 2003. PKC 2003. Lecture Notes in Computer Science, vol 2567. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-36288-6_23.

[10] Bagaria, V., Dembo, A., Kannan, S., Oh, S., Tse, D., Viswanath, P., ... & Zeitouni, O. (2022, November). Proof-of-stake longest chain protocols: Security vs predictability. In Proceedings of the 2022 ACM Workshop on Developments in Consensus (pp. 29-42).