Chapter 3
Wallet Security

Carlo Parisi, Dyma Budorin, and Ostap Khalavka

3.1 Introduction

3.1.1 Overview of Different Types of Blockchain Wallets

Blockchain wallets act as storage for users' digital assets. Cryptocurrency assets are as safe as the wallets they are stored in. Through blockchain wallets, users interact with the blockchain network. There are two main types of blockchain wallets, including software and hardware wallets, which are also referred to as "hot" and "cold" storage. In reality, blockchain wallets do not store cryptocurrencies. Instead, they generate the information needed to send and receive tokens. This information includes public and private keys.

3.1.2 Explanation of the Different Kinds of Wallets

Cold wallets and hot wallets are types of cryptocurrency wallets used for storing digital assets. Cold wallets, also known as offline or hardware wallets, are not connected to the Internet, providing higher security for long-term storage. Examples include the Ledger Nano S and paper wallets. Hot wallets, also known as online or software wallets, are connected to the Internet and are more convenient for frequent transactions. Examples include Exodus, Mycelium, and MetaMask. Cold wallets are less susceptible to hacks and cyberattacks, while hot wallets are more vulnerable due to their constant Internet connection. The choice between a cold wallet and a

C. Parisi (✉) · D. Budorin · O. Khalavka
Hacken, Lisbon, Portugal
e-mail: c.parisi@hacken.io; d.budorin@hacken.io; o.khalavka@hacken.io

K. Huang et al. (eds.), *A Comprehensive Guide for Web3 Security*, Future of
Business and Finance, https://doi.org/10.1007/978-3-031-39288-7_3

hot wallet depends on the user’s priorities, whether it is long-term asset security or ease of access for daily transactions.

Blockchain wallets may be further divided into four big categories depending on the device or software used for their management.

Mobile wallets: These are suitable for users who frequently use cryptocurrencies for payments. Mobile wallets are apps that store users’ private keys and often use Simplified Payment Verification (SPV) technology, which operates on smaller subsets of the blockchain and relies on trusted nodes in the network (Potapenko et al., 2021).

Web wallets: These store users’ private keys, making them more susceptible to hacks and third-party collapses. It is essential to choose a reputable web wallet provider with robust security measures in place.

Desktop wallets: Once downloaded and installed on a user’s computer, desktop wallets store private keys on the user’s hard drive or solid-state drive.

Hardware wallets: These store private keys on a secure physical device, making them resistant to computer viruses. Hardware wallets are ideal for users who prioritize the long-term storage and protection of their assets.

The choice of wallet type depends on the user’s digital asset management strategy. A combination of wallet type could be used, hardware wallets for assets that should be more safe and less accessible and other kinds of wallets for assets that should be more easily accessible (Sharma, 2023).

3.1.3 Comparison of Blockchain Wallets to Traditional Banking and Financial Systems

The traditional banking system does not eliminate major issues attributable to transactions. Namely, transactions may be slow and have to pass through an intermediary. The traditional banking system works on the Internet and uses its own software, while blockchain wallets work on the blockchain. Blockchain wallets do not process fiat money, and only crypto serves as a medium of exchange.

Unlike traditional banking, *blockchain wallets do not need centralized command to process every transaction*, and, thus, users are the only ones who can determine whether to conduct a transaction. *The other thing about blockchain wallets is that they cannot be frozen by outside parties*. In traditional banking, financial institutions can freeze a client’s account or block any transactions at the request of law enforcement authorities. *Blockchain wallets do not let any party interfere with the user–network interaction chain*.

Geographical location has no bearing on blockchain wallet operations. The speed of processing does not depend on the physical distance between the cryptographic sender and receiver. The functioning of blockchain wallets is governed by smart contracts, which automate agreements. Only when certain conditions of the agreement are met does the execution of a transaction take place. Unlike in the world of blockchain, in the financial world, transactions take more time and cost more since they are based on greater trust, requiring manuals and paper to meet legal rules and avoid possible implications.

3.1.4 Difference between Custodial and Non-custodial Wallets

There are also two broad categories of digital crypto wallets, depending on who has full control over the user’s assets (Academy, 2023).

Custodial wallets: There is a third party holding and managing a private key to the user’s wallet on his or her behalf and holding his or her assets in custody. An example of a custodial wallet is an account on a centralized crypto exchange. Even if a customer forgets or loses his or her cryptocurrency exchange account password, he or she will still be able to access the account and its assets by contacting a third-party customer support service.

Non-custodial wallets: Users alone have complete control over their assets. Non-custodial wallets are a good option for experienced users who know how to safely store their private keys and secret phrases. Non-custodial wallets are used for interactions with decentralized exchanges and decentralized applications. One of the best things about non-custodial wallets is that they do not charge a custodial fee. On the other hand, *the user has to take more responsibility for managing a non-custodial wallet*.

3.2 How Blockchain Wallets Work

3.2.1 Technical Explanation of How Blockchain Wallet Works

There are two main types of wallets in the crypto space: nondeterministic and deterministic.

Nondeterministic wallets, which are also called “Just a Bunch of Keys” (JBOK) wallets, make each key from a different random number. These keys have nothing in common.

On the other hand, deterministic wallets generate all keys from a single master key, known as the seed. All the keys in this type of wallet are related to each other and can be regenerated if the original seed is known (Antonopoulos, 2017). Deterministic wallets often use a hierarchical structure, like in BIP-32 (B, 2022) and BIP-44 (B, 2019).

For added security against data loss, deterministic wallets often use a mnemonic code made up of a list of words. This can be written down and used in the event of an accident, such as losing your phone. However, *it is important to keep these code*

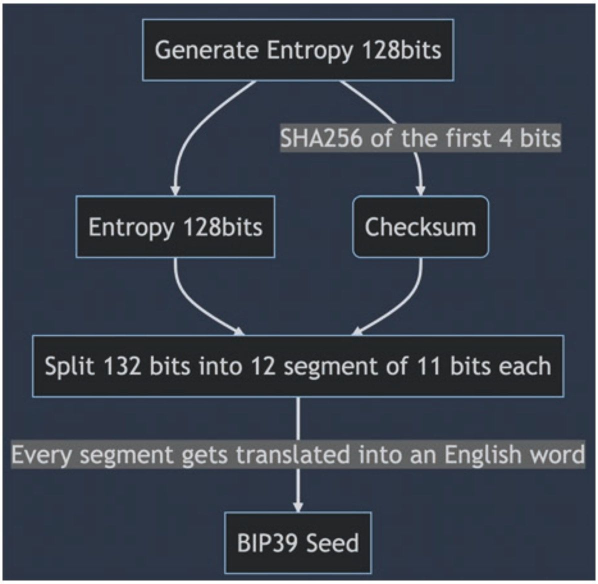
Generate Entropy 128bits
SHA256 of the first 4 bits
Entropy 128bits
Checksum
Split 132 bits into 12 segment of 11 bits each
Every segment gets translated into an English word
BIP39 Seed

Fig. 3.1 From Entropy to BIP39 Seed

words safe, as someone with access to them can recreate your wallet and gain access to your funds. It is recommended to write them down on paper and store them in a secure location, *never storing them electronically*.

The mnemonic phrase, which is also called the “seed phrase” or “recovery phrase,” is a set of words that are made using a standard method described in BIP-39 (B, 2020) (Fig. 3.1).

The process starts by creating a cryptographically random sequence of 128 to 256 bits, known as S. A checksum is then added to this sequence by taking the first (length-of-S, 32) bits of the SHA-256 hash of S. The checksum is then appended to the end of the random sequence S. This sequence and checksum are then divided into groups of 11 bits, and each group is mapped to a word from a predefined dictionary of 2048 words. When you put these words together in the order they were made, you get the final mnemonic phrase, which is a unique, easy-to-remember phrase that can be used to restore a wallet.

3.2.2 Overview of the Use of Public and Private Key

The process of generating a private key or master key on a blockchain can vary. In a Bitcoin wallet, for example, there is a set of key pairs, each of which is made up of a private key and a public key. The private key is a randomly chosen number. Elliptic curve multiplication, which is a one-way cryptographic function, is used on the private key to make the public key. The Bitcoin address is then created by using a one-way cryptographic hash function on the public key (Antonopoulos, 2017).

In Ethereum, the process is slightly different. There are two types of accounts: Externally Owned Addresses (EOAs) and contracts. EOAs have a public and private key pair similar to Bitcoin. Contract addresses are made when a special transaction is sent to the 0 address (20 bytes of 0s with 0x as a prefix), and the address is based on the sender's public address and their nonce (Antonopoulos & Wood, 2021).

Digital signatures, which are made with the private key, are used to get access to and control over funds in a blockchain. Transactions need a valid digital signature in order to be included in the blockchain. Anyone who knows how to get a private key can control the account and any coins or tokens that go with it. *As long as a user keeps their private key safe, digital signatures in Ethereum transactions confirm the true owner of the funds by proving ownership of the private key.*

The process of generating a public key from a private key in Bitcoin is done through elliptic curve multiplication. The private key, which is shown as a random number k, is multiplied by the generator point G, a fixed point on the curve. This results in another point on the curve, which is the corresponding public key K. The secp256k1 standard describes the generator point, which is the same for all Bitcoin keys. The equation can be represented as:

$$K = k * G \tag{3.1}$$

Generating a public key (K) from a private key (k)where k is the private key, G is the generator point, and K is the resulting public key. *Since the generator point is the same for all users, the same private key multiplied by G will always result in the same public key.* The relationship between the private key and the public key is fixed and can only be calculated in one direction, from the private key to the public key (Antonopoulos & Wood, 2021; Antonopoulos, 2017). This is why you can share a Bitcoin address, which is based on the public key, without giving away your private key.

Using a one-way cryptographic hashing method, a Bitcoin address can be made from a public key. This process involves applying a "hash algorithm," a one-way function that produces a unique fingerprint or "hash" of an input of any size. Cryptographic hash functions are used extensively in Bitcoin, including in the creation of Bitcoin addresses, script addresses, and the mining Proof-of-Work algorithm. The specific hash algorithms used to generate a Bitcoin address from a public key are SHA-256 and RIPEMD-160. To create a Bitcoin address, the public key K is first hashed using SHA-256, and then the result is hashed again using RIPEMD-160, resulting in a 160-bit (20-byte) number (Fig. 3.2).

$$A = RIPEMD160(SHA256(K))$$

Fig. 3.2 Generating a Bitcoin Address (A) from a private key (K)

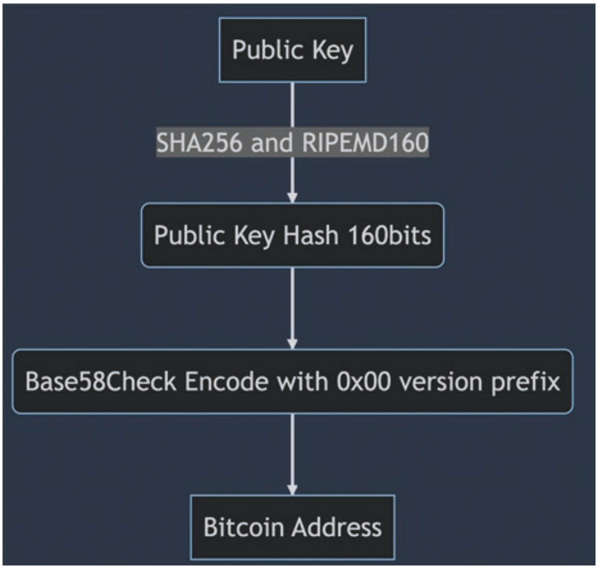

Fig. 3.3 From public key to bitcoin address

$$A(pr) = \beta 96 \ldots 255(KEC(ECSDAPUBKEY(pr)))$$

Fig. 3.4 Generating a public/private key pair in Ethereum

Bitcoin addresses are usually encoded with Base58Check, which uses 58 characters and a checksum to make them easier to read, make them less confusing, and prevent mistakes when they are written down or typed in (Antonopoulos, 2017) (Fig. 3.3).

In Ethereum, the process of generating a public/private key pair is as follows (Fig. 3.4):

The corresponding Ethereum address, A(pr), can be found by taking the 160 bits on the right of the Keccak hash of the corresponding ECDSA public key. This is done with a private key, pr. To put it simply, the process of getting a public key from a private key is similar to the process used in Bitcoin, where $K = k*G$. The public key is a 64-byte string of letters and numbers. The public address used to receive transactions can be made from the public key by using a one-way hash function (Keccak-256) to turn it into a 32-byte string, then taking the last 20 bytes and adding a 0x prefix to them. When it comes to contracts in Ethereum, there are two methods for deriving the address: the opcode CREATE and the opcode CREATE2 proposed in the Ethereum Improvement Proposal (EIP) 1014 (Proposals, 2018). The most common method is CREATE, which gets the address by taking the rightmost 160 bits of the Keccak hash of the RLP encoding of a structure that contains the sender address and the account nonce. The sender is the address that initiates the

$$\alpha = \beta 96 \ldots 255(KEC(RPL((s, \sigma[S]n - 1)))$$

Fig. 3.5 Generating a contract address with the CREATE opcode

transaction, and the nonce is a counter that tracks the number of transactions sent by an account (Antonopoulos & Wood, 2021) (Fig. 3.5).

3.2.3 Overview of the Role of the Wallet in Public and Private Key

A wallet is a piece of software that connects you to the blockchain and is mostly used to keep track of keys and addresses. It makes a private key by picking a number between 1 and 2**256 at random. It does this by using a good source of entropy to make sure the number is not predictable or deterministic. As was already said, the public key and address can be figured out from the private key. The private key is used to authorize transactions and show ownership of funds, while the public key is used to get the address that is used to receive transactions or funds. Public keys and addresses can be shared publicly because the process used to derive them is one-way, and *it is computationally infeasible to determine a private key from a public key*.

3.2.4 Smart Contract Wallets

A smart contract wallet is a type of wallet that combines the benefits of custodial and non-custodial wallets. With custodial wallets, there is a third party who can withdraw funds from the wallet if he so desires, whereas with non-custodial wallets, the user bears a significant amount of responsibility for the security of the private key (DeCommas, 2022).

Generally, a smart contract wallet is a smart contract that acts as a wallet through account abstraction; the most famous example of a smart contract wallet are multisignature wallets.

3.2.5 Account Abstraction

There are currently three major challenges facing the widespread adoption of blockchain technology:

1. The complexity of the technology and the difficulty of providing a user-friendly experience
2. The management of wallets and seed phrases by companies and escrow services
3. The recovery of lost or stolen wallets due to poor operational security

Even though these problems have not been fully solved yet, many people use account abstraction (Julien Niset, 2022), multisignature wallets, and social recovery methods to deal with them.

Account abstraction aims to merge the two types of Ethereum accounts, externally owned accounts and contract accounts, into one unified contract account (Team, 2023). Transactions will also move from the blockchain to the Ethereum Virtual Machine (EVM), eliminating the need for separate account types. The main goal of this change is to make things easier for users by letting developers make better protocols and services without having to think about different types of accounts. Additionally, it will offer advanced features such as multisignature security, social recovery, rate limiting, and gasless meta-transactions.

There are currently various EIPs that aim to improve account abstraction. These include EIP-86 (Vitalik Buterin, 2017), EIP-2938 (Proposals, 2020a), EIP-3074 (Proposals, 2020b), and EIP-4337 (Proposals, 2021), which are notable proposals in this area.

When Ethereum account abstraction is finished, it will change how accounts are implemented and how users interact with them. It will also give developers the freedom to create and manage accounts however they want.

With account abstraction, developers will be able to utilize smart contract logic not just for determining transaction effects but also for fee payment and validation. This will provide important security benefits like multi-sig and smart recovery wallets and the ability to change keys without switching wallets.

Some of the use cases that account abstraction will make possible include:

Wallets: With account abstraction, users will be able to enjoy advanced security features such as multi-sig and smart recovery, as well as the convenience of changing keys without changing wallets.

Sponsored transactions: Account abstraction will let entities or their subsidiaries do things like pay fees on behalf of users and let users pay gas fees in ERC-20 tokens, which will be turned into ETH.

Meta-transactions: With account abstraction, users can receive meta-transactions (gasless) and pay for gas without having to trust a relayer.

3.2.6 Multisignature Wallets

Bitcoin is usually stored in a single-key address, which means that only the person who has the private key to that address can use the money. This means that only one key is needed to sign transactions, and anyone with the private key can transfer the coins without any authorization. *However, this system presents security issues as it* *is vulnerable to phishing attacks, and the funds are protected by a single point of failure*. Additionally, this method is not ideal for businesses, as the private key would either be entrusted to a single person or multiple individuals, which is not secure.

Multi-sig wallets solve these problems by making it so that you need more than one signature, *made with different private keys*, to get to the money on an address. An example for a configuration is 2-of-3, which means that you only need two signatures to get to the money in a 3-signature address.

Multi-sig technology has a variety of potential applications; some of the most common use cases include:

- Corporate treasury management
- Escrow services
- Secure storage of digital assets
- Crowdfunding platforms
- Secure transfer of sensitive information

It is a more secure way of handling cryptocurrency funds and can be used for a variety of purposes, such as corporate treasury management, escrow services, and digital asset storage.

3.2.7 Social Recovery

To make sure a cryptocurrency wallet is safe, it should be made in a way that meets three key criteria:

1. No single point of failure: The wallet should not have a single point of vulnerability that can be exploited by an attacker to gain access to funds or a single point of loss that can deny access to funds.
2. Low mental overhead: The design should be user-friendly and not require users to learn new habits or exert mental effort to follow specific patterns of behavior.
3. Maximum ease of transacting: Normal activities such as transactions should not require much more effort than traditional wallets like Status or MetaMask.

One method that is gaining popularity for securing a wallet is social recovery. In this system, there is a single "signing key" that can be used to approve transactions.

There is a set of at least three "guardians" (or a higher number), and a majority of them can cooperate to change the signing key of the account.

The signing key has the ability to add or remove guardians, but only after a delay (often 1–3 days).

This method allows for a secure and user-friendly way to protect the wallet while also allowing for flexibility in case of loss or theft of the private key (Fig. 3.6).

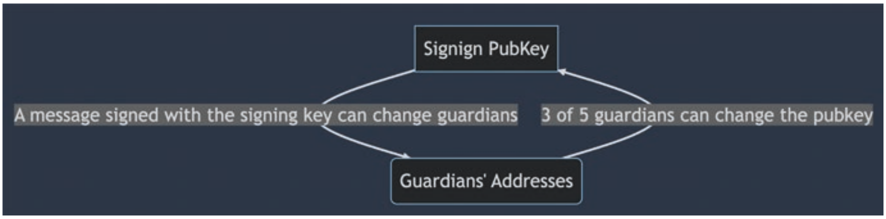
Signign PubKey
A message signed with the signing key can change guardians
3 of 5 guardians can change the pubkey
Guardians' Addresses

Fig. 3.6 *Signing Pubkey* and *Guardians' Addresses* correlation

3.2.8 MPC Wallet

MPC, or secure Multi-Party Computation, is a branch of cryptography that has been around for 30 years. It enables two or more parties to jointly compute the output of a function, without revealing their respective inputs.

For cryptocurrency wallets, MPC allows for the creation of a secure key management system without a single point of failure. Multiple parties, such as a mobile phone and a remote server, can perform cryptographic functions like key generation and transaction signatures together, while keeping their secrets confidential. In this process, there is never a single private key that is generated, split, or reconstructed (Portier & Diya, 2023; Wikipedia contributors, 2023; ZenGo, 2023).

By using MPC technology, wallets, both for consumers and institutions, can create a secure on-chain asset management system without a single private key. *This eliminates the risk of private key theft and key loss, as each party can individually back up their secret input without exposing the entire system.*

The familiarity and ease of using recovery methods such as email, trusted contacts, cloud backup, or biometric scanning make them less intimidating for most people. This is key to attracting new users to the world of crypto.

Having recovery options that are familiar to people will help them feel more comfortable using crypto. Once they are in the ecosystem, they may opt for more secure options or low-centralization risk alternatives.

Using multiple wallets for storing crypto assets is recommended. However, the biggest barrier for most people entering the crypto world is the use of seed phrases.

> The idea of having a single phrase that controls the entire contents of an account can be daunting for many people. While some are willing to take full responsibility for keeping it safe, most are not.

Creating an easy entry point for new users to try crypto applications and hold assets without the worry of seed phrases is critical for attracting the next billion people to the world of Web3.

3.2.9 *Most Common Attack Vectors*

Attackers use a diversified portfolio of attack techniques to compromise blockchain wallets. Generally, these malicious techniques may be divided into the following broad categories:

Stealing private keys: Keys must be encrypted at the application level. Hackers can easily steal unencrypted keys in the application sandbox, clipboard, preference or external areas, SD card, etc.

Malicious devices: Hackers can gain access to the client's blockchain address by abusing common software development tools or if a device is rooted or jailbroken.

Man-in-the-Middle attacks: Malicious actors infiltrate a conversation between a user and an application to impersonate one of these parties. One of the most common ways for users to fell victim to man-in-the-middle attacks is to use unsecure public Wi-Fi networks.

Malware: Malicious actors actively utilize ads-delivered malware to drain victims' wallets. Instead of clicking on the official website, unsuspecting victims click on sponsored advertisements. This form of malware used by attackers is information-stealer that is often hidden on highly convincing phishing pages.

Phishing: Malicious techniques aimed at stealing private keys and security recovery phrases Attackers create websites or services that appear to be genuine, such as a service to recover stolen funds. After gaining all sensitive data, an attacker can access someone's crypto wallet and exfiltrate assets. Phishing attacks are frequently disguised as attractive emails that appear to be legitimate. All these suspicious emails have the same purpose: to lure users into sharing some sensitive data or into performing actions that would lead to the loss of funds or the disclosure of data, including seed phrases (Pirus, 2020).

Threatful browser extensions: People like using browser extensions such as print screens, grammar checks, etc. However, apart from simplifying the user's life, some of these extensions, especially the ones coming from suspicious sources, may monitor and copy user's data and transfer them to hackers. That is why users should not install unverified browser extensions.

3.2.10 *Wallet Security Features*

Wallets for cryptocurrencies usually have more than one way to keep the user's money and private information safe. The following are a few of the most popular security features offered by cryptocurrency wallets:

- Private key management is essential to a crypto wallet since it allows users to sign transactions and gain access to their money. The majority of wallets provide solutions for managing private keys, such as hardware wallets that store the private key offline or multi-sig wallets that need several signatures to approve a transaction.

- By requiring the user to provide two forms of authentication, such as a password and a one-time code delivered to their phone, in order to access their funds, two-factor authentication (2FA) increases security.
- Cryptocurrency wallets frequently employ encryption to safeguard private keys and other sensitive data, as well as to block unauthorized access even in the event that a device or computer is lost or stolen.
- Good wallets provide backup and recovery alternatives, such as seed phrases, to help users get their money back if they lose access to their wallet.
- Software upgrades on a regular basis can shield users against threats and address any faults in the wallet program. Attack vectors are occasionally only identified years after the wallet has been released, as was the case with the fault-injection technique identified for the Trezor hardware wallet, which was subsequently patched through software updates (Zetter, 2022).

The user is ultimately the biggest weakness for the majority of wallets. So, a major security feature, along with enough randomness for key generation, strong encryption to keep the keys safe, and ways to fix software or hardware problems, is the ability to create an interface that connects the user to the blockchain in a way that is easy to use and gives them useful information.

It's important to choose a wallet with strong security features, and it's also a good idea to keep private keys and seed phrases safe, back up the wallet regularly, and keep up with the blockchain while working with it.

3.3 Past Blockchain Wallet Hacks

3.3.1 Overview of Past Relevant Wallet Hacks

3.3.1.1 "NFT God" Hacked, January 2023

On January 14, 2023, "NFT God" (real name Alex) downloaded the video streaming service OBS, but instead of the original link, he used a sponsored link on Google that contained malicious software. A few hours later, one of God's followers alerted him that his Twitter account had been hacked. However, the account compromise was only the first in a series of attacks. All wallets belonging to "NFT God" were drained of crypto and NFTs. The attacker also hacked Alex's Gmail, Discord, and Substack accounts and sent emails containing malicious links to more than 16K subscribers (Connor Sephton, 2023).

Alex made a costly mistake when setting up his ledger account. According to him, he entered his seed phrase in a way that no longer kept the wallet cold. Because Alex made this mistake, the hacker was able to use malware to get into his wallet.

3.3.1.2 BitKeep, December 26 2022, $8M Lost

On December 26, 2022, some users of the BitKeep wallet reported seeing their assets drained from their wallets. The team confirmed that the attacks hijacked some Android package (APK) downloads and installed them with code. APK is the file format that allows users to install apps from third-party sources on their Android phones. This incident cost users around $8 million.

In October of the same year, BitKeep also experienced an exploit, during which the attacker took $1 million worth of BNB through the service enabling token swaps (Reguerra, 2022).

3.3.1.3 Deribit, November 2022, $28M Lost

On November 2, 2022, Deribit cryptocurrency options and futures exchange informed its community of the compromise of its hot wallets. Hackers managed to get access to the exchange's hot wallet and initiate withdrawals. The attack affected hot wallets for Bitcoin, Ethereum, and USDC. The company's officials noted that 99% of all funds were stored in cold storage, which is why the incident did not have catastrophic implications for the exchange (Knight, 2022).

3.3.1.4 Solana Wallet Hacks, August 2022, $5M Lost

The attack affected nearly 8000 Solana digital wallets. The incident affected major Internet-connected "hot" wallets such as Phantom, Slope, and TrustWallet. The attack predominantly affected mobile wallet users. The attacker managed to sign transactions on users' behalf. The trusted third-party service might have been compromised through a supply-chain attack. The bug that was used by attackers was likely in the software that several software wallets used (Quarmby, 2022).

3.3.1.5 Profanity Wallets Hack, $3.3M Lost

Throughout 2022, Profanity wallets might have experienced hacks due to the ambiguity in the creation of vanity addresses. Profanity is the tool allowing users to create "vanity addresses"—custom crypto wallets that have identifiable names or numbers within them. Security researchers found out that the generator seeded 256-bit private keys with a random 32-bit vector. Profanity seeded the cryptographic pseudorandom number generator with an unsigned integer, thereby leaving only 4.3 billion seed values possible. Although this figure is great, it is not sufficient to let crypto wallets withstand a brute-force attack. In this case, users should be aware of the importance of using reputable and still actively supported tools for private key generation. The original creators of the Profanity vulnerability address generator abandoned the product multiple years ago, but, unfortunately, users still turn to this tool for private key generation (Emmanuel, 2022).

3.3.1.6 Binance Hot Wallet Hack, May 2019, $40M Lost

In May 2019, hackers broke into the Binance exchange and stole 7000 bitcoins. One of the biggest cryptocurrency exchanges, Binance, later paid back the losses using the SAFU (Secure Asset Fund for Users), a separate fund set up to protect users' money in case of theft.

Combining phishing, malware, and other methods, the breach gave the attackers access to the private keys for Binance's hot wallet. Even though Binance tried hard to keep its platform safe, the hackers were able to pull off the heist (Doug Bonderud, 2019).

3.3.1.7 MetaMask iCloud Hack, July 2021

In July 2021, there was an event known as the MetaMask iCloud hack, during which a hacker gained access to a user's MetaMask seed phrase that was kept in their iCloud account. The user's Ethereum assets that were kept in the MetaMask wallet were now accessible to the hacker. The attack happened because the user did not protect their iCloud account and MetaMask seed phrase well enough.

Not a bug in the MetaMask software, but the user's failure to follow best practices for storing and protecting their seed phrase made this type of attack possible. To avoid situations like this one, you should always save your seed phrase in a safe place, avoid storing it on cloud-based services, and turn on two-factor authentication for all of your online accounts (Toulas, 2022).

3.3.1.8 Parity Multisig Hack, November 2017, $30M Lost

In November 2017, there was a security breach called the Parity Wallet hack. About 153,000 ETH, which was worth about $30 million at the time, were stolen from multi-sig wallets made with Parity Technologies' Ethereum client software. A weakness in the wallet contract code made it possible for the hacker to take over the contract owner of the wallet and make transactions from it. Users who were hurt by the incident lost a lot of money, and the incident made people worry about how safe smart contract systems are. In response to the event, Parity Technologies tried to pay the users who were hurt and make its products safer (Palladino, 2020).

3.3.2 The Impact of Cyberattacks on Blockchain Wallets

Because there are so many ways to attack blockchain wallets, the projects that make them have had to add more security measures. Following a series of hot wallet compromises, exchange wallets establish special insurance funds to immediately cover potential losses. At the same time, hot wallet hacks force crypto exchanges to hold the majority of the assets they manage in cold wallets.

Mobile wallet developers implement security protections such as limiting the wallet's functionality on jailbroken or rooted devices, limiting the device's lifecycle, and ceasing support for old devices.

Also, it is a common mistake to suggest that crypto wallet developers are the best experts in cryptography. Most of the time, they are just regular web, mobile, and desktop developers who know the basics of cryptography but are not experts in how it is used. That is why the companies that develop blockchain wallets invest heavily in the education of their employees to make them more advanced specialists in cryptography.

Early detection of bugs could be critical for crime prevention. As a result, blockchain wallet developers begin to pay more attention to communication with users about flaws they notice when using their wallets.

And, in general, using best practices for secure coding, like OWASP Secure Coding Practices, is the best way to protect their products from the most common types of attacks.

3.4 The Importance of Auditing a Wallet

3.4.1 Explanation of the Importance of Auditing a Wallet

The users' trust in crypto wallets primarily depends on their security. From the developers' perspective, the crypto wallet attack area is enormous, while attackers may need to apply just a single attack vector to reach their malicious targets. Regarding the rapid speed of crypto transactions, security flaws in crypto wallets allow attackers to drain money quickly without even allowing developers to notice the attack and respond appropriately. Also, public blockchains do not have support services that can revert transactions or anti-fraud systems to timely notify a user of the susceptibility of certain actions. That is why stopping an ongoing attack targeting crypto wallets does not always bring the desired outcomes and is likely to result in an ultimate failure. For crypto wallets, attack prevention is a reasonable strategy for developers. *The most effective way to stop attacks is to audit a wallet to find bugs and fix them before attackers can see them.*

The other important reason for blockchain wallet developers to turn to security testing is related to budgeting. Pushing security from the earliest stages prevents spending on fixing exploitation outcomes and dealing with reputation damage.

Investing in security from the start saves projects valuable time that would otherwise be spent recovering wallets after an attack.

Just putting information security controls in place is not enough. Professional auditors are the only ones who can check if defenses are set up correctly and if there are any vulnerabilities that have not been found yet. Security testing of crypto wallets shows their developers what components of their products may be subject to a cyberattack, as well as pointing out expected attack scenarios.

Professional auditors not only look at specific flaws but also assess their synergies. The purpose of auditing a wallet is to ensure the wallet's cryptographic confidentiality, integrity, and availability, as well as the cryptographic assets it stores and its private keys.

Depending on the type of wallet under test and its technical peculiarities, auditors identify specific attack vectors and advise on the measures to be taken to eliminate these threats or minimize their business impact on a project.

Overall, security testing gives users more faith in their crypto wallets by showing them that their assets are well protected and that there is no chance of unauthorized interference.

3.4.2 Overview of Different Types of Audit that Can Be Performed

Crypto wallet security testing has several main forms, depending on the technical peculiarities of the product to be audited:

Threat modeling and risk assessments: This form of testing identifies the threats affecting the wallet under audit that may arise from different sources such as the attacks by cybercriminals, insider attacks, etc. The purpose of this form of testing is to assess the threats depending on their likelihood and potential impact so that a project can set priorities for their proper management and mitigation.

Penetration testing: Emulation of attack methods done by hackers to get into the targeted crypto wallet to identify vulnerabilities and point out the measures to be taken by a project to fix these flaws.

Architecture review: Validation of the presence and viability of implemented security controls. The purpose of this form of testing is to highlight any possible security risks in the wallet's architecture.

Code review: Thorough analysis of the code quality to detect incorrect functions and any flaws in dependencies that would undermine wallet's security. A special focus during code review is made on signing transactions and key generation.

The complex blockchain wallet audit process also includes identity management audits, certification and authorization audits, node security reviews, session management checks, and cryptography security audits.

3.4.3 Different Tools to Audit Wallets

Auditing a blockchain wallet, which includes cryptographic-using browser add-ons, desktop applications, hardware, and more, is a difficult operation. I*t's crucial to evaluate both the cryptography part and the application element while auditing a wallet*.

Depending on the type of wallet being utilized, the application side of the wallet will differ substantially. For instance, it is important to confirm that an extension does not have any communication flaws with the browser or the Internet at large while auditing a browser wallet. This can entail performing an HTTP response header security audit, an XSS security audit, and a third-party JS assessment. It may be crucial to look at the app cache security or permission detection while auditing a desktop wallet.

As hardware and software have been audited since the invention of computers, these audits are not specific to wallets. Combining these audits with the cryptography component makes auditing wallets complicated. *A wallet must meet a number of requirements in order to be considered secure, such as producing or receiving enough entropy while generating keys*. Ensuring that your system complies with NIST SP 800-90A is one approach to doing this. A cryptographic security audit is typically required for a wallet to be deemed secure.

Some of the instruments used to conduct a cryptographic security audit include the following:

CryptCheck: A program to evaluate the effectiveness of SSL/TLS setups, including those for ciphers and certificates.

SSL Labs: An online service called SSL Labs offers a thorough examination of SSL/TLS setups and certificates.

Qualys SSL Labs: A web-based application that offers a thorough study of SSL/TLS implementations.

OpenSSL: A popular open-source SSL/TLS implementation that has a number of tools for testing and debugging encryption setups.

References

Academy, B. (2023, February 9). *Custodial vs. Non-custodial wallets: What's the difference?* Binance Academy. Retrieved from https://academy.binance.com/en/articles/custodial-vs-non-custodial-wallets-what-s-the-difference

Antonopoulos, A. M. (2017). *Mastering bitcoin: Programming the open Blockchain*. O'Reilly Media.

Antonopoulos, A. M., & Wood, G. (2021). *Mastering Ethereum: Building smart contracts and DApps: Building smart contracts and DApps*.

B. (2019, March 12). *bips/bip-0044.mediawiki at master · bitcoin/bips*. GitHub. Retrieved from https://github.com/bitcoin/bips/blob/master/bip-0044.mediawiki

B. (2020, December 20). *bips/bip-0039-wordlists.md at master · bitcoin/bips*. GitHub. Retrieved from https://github.com/bitcoin/bips/blob/master/bip-0039/bip-0039-wordlists.md

B. (2022, January 3). *bips/bip-0032.mediawiki at master · bitcoin/bips*. GitHub. Retrieved from https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki

Bonderud, D. (2019, May 15). *Binance hack steals $41 million from 'hot wallet'*. Security Intelligence. Retrieved from https://securityintelligence.com/news/binance-hack-steals-41-million-from-hot-wallet

Buterin, V. (2017, February 10). *EIPs/eip-86.md at master · ethereum/EIPs*. GitHub. Retrieved from https://github.com/ethereum/EIPs/blob/master/EIPS/eip-86.md

Connor Sephton. (2023, January 16). *"Violated": NFT god loses "life-changing" sum of Crypto after clicking on Malware Link*. CoinMarketCap Alexandria. Retrieved from https://coinmarketcap.com/alexandria/article/violated-nft-god-loses-life-changing-sum-of-crypto-after-clicking-on-malware-link

DeCommas. (2022, November 8) *Smart contract wallets explained*. Retrieved from https://decommas.io/blog/smart-contract-wallets-explained

Emmanuel, O. O. (2022, September 26). *Hacker exploits profanity's vanity address to steal $950 in ETH*. crypto.news. Retrieved from https://crypto.news/hacker-exploits-profanitys-vanity-address-to-steal-950-in-eth/

Knight, O. (2022, November 2). *Crypto exchange Deribit loses $28M in hot wallet hack, pauses withdrawals*. Retrieved from https://www.coindesk.com/business/2022/11/02/crypto-exchange-deribit-loses-28m-in-hot-wallet-hack/

Niset, J. (2022, March 28). *Part I: WTF is account abstraction*. argent.xyz. Retrieved from https://www.argent.xyz/blog/wtf-is-account-abstraction

Palladino, S. (2020, May 19). *The parity wallet hack explained*. OpenZeppelin Blog. Retrieved from https://blog.openzeppelin.com/on-the-parity-wallet-multisig-hack-405a8c12e8f7

Pirus, B. (2020, September 6). *Electrum Bitcoin wallet still plagued by known crypto phishing attack*. Cointelegraph. Retrieved from https://cointelegraph.com/news/electrum-bitcoin-wallet-still-plagued-by-known-crypto-phishing-attack

Portier, B., & Diya, C. (2023, January 25). *How confidential space and MPC can help secure digital assets*. Google Cloud Blog. Retrieved from https://cloud.google.com/blog/products/identity-security/how-confidential-space-and-mpc-can-help-secure-digital-assets.

Potapenko, J., Hil, A., & Voitova, A. (2021, December 13). *Crypto wallets security as seen by security engineers*. Cossack Labs. Retrieved from https://www.cossacklabs.com/blog/crypto-wallets-security.

Proposals, E. I. (2018, April 20). *EIP-1014: Skinny CREATE2*. Ethereum improvement proposals. Retrieved from https://eips.ethereum.org/EIPS/eip-1014

Proposals, E. I. (2020a, September 4). *EIP-2938: Account abstraction [DRAFT]*. Ethereum improvement proposals. Retrieved from https://eips.ethereum.org/EIPS/eip-2938

Proposals, E. I. (2020b, October 15). *EIP-3074: AUTH and AUTHCALL opcodes [DRAFT]*. Ethereum Improvement proposals. Retrieved from https://eips.ethereum.org/EIPS/eip-3074

Proposals, E. I. (2021, September 29). *ERC-4337: Account abstraction using Alt Mempool [DRAFT]*. Ethereum Improvement Proposals. Retrieved from https://eips.ethereum.org/EIPS/eip-4337

Quarmby, B. (2022, August 3). *Solana-based wallet hack saw millions drained*. Cointelegraph. Retrieved from https://cointelegraph.com/news/ongoing-solana-based-wallet-hack-has-already-seen-millions-drained.

Reguerra, E. (2022, December 26). *Hackers drain $8M in assets from Bitkeep wallets in latest DeFi exploit*. Cointelegraph. Retrieved from https://cointelegraph.com/news/hackers-drain-8m-in-assets-from-bitkeep-wallets-in-latest-defi-exploit

Sharma, T. K. (2023, January 25). *Types of crypto wallets explained*. Blockchain Council. Retrieved from https://www.blockchain-council.org/blockchain/types-of-crypto-wallets-explained/.

Team, P. (2023, March 28). *Ethereum account abstraction: Everything you need to know!* Panther Protocol Blog. Retrieved from https://blog.pantherprotocol.io/ethereum-account-abstraction-everything-you-need-to-know

Toulas, B. (2022, April 18). *Hackers steal $655K after picking MetaMask seed from iCloud backup*. BleepingComputer. Retrieved from https://www.bleepingcomputer.com/news/security/hackers-steal-655k-after-picking-metamask-seed-from-icloud-backup

Wikipedia contributors. (2023, March 27). *Secure multi-party computation*. Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Secure_multi-party_computation.

ZenGo. (2023, March 30). *MPC wallet - What is MPC? - ZenGo*. Retrieved from https://zengo.com/mpc-wallet

Zetter, K. (2022, January 24). *Cracking a $2 million crypto wallet*. The Verge. Retrieved from https://www.theverge.com/2022/1/24/22898712/crypto-hardware-wallet-hacking-lost-bitcoin-ethereum-nft