




# Where Are My Cryptos?

Miquel Calonge<sup>1</sup>, Edgar Batista<sup>1</sup>, Julio Henández-Castro<sup>2</sup>,  
and Agusti Solanas<sup>1</sup> 

<sup>1</sup> Universitat Rovira i Virgili, Tarragona, Spain  
`miquel.calonge@urv.cat`, `edgar.batista@urv.cat`, `agusti.solanas@urv.cat`  
<sup>2</sup> University of Kent, School of Computing, Canterbury, UK  
`j.c.hernandez-castro@kent.ac.uk`

**Abstract.** The financial sector has suffered a groundbreaking transformation with the advent of cryptocurrencies, shifting from centralised to decentralised schemes. Hardware wallets play an essential role in storing cryptocurrencies securely. However, these electronic devices generally have limited resources that open the door to attacks. In this article, we describe three attacks against them. Several wallets with funds or recent transactions have been discovered with these attacks.

**Keywords:** Hardware wallets · Cryptocurrencies · Entropy · Derivation paths · Seed phrases · Security

## 1 Introduction

In recent years, the financial landscape has undergone a revolutionary transformation with the emergence of cryptocurrencies [1]. One of the most significant changes this digital revolution brings is the newfound responsibility placed on users to safeguard their assets. Eliminating intermediaries, such as banks and other financial institutions, implies users have absolute control over their digital wealth. This decentralisation eradicates the single point of failure that could expose funds to potential hacks. Blockchain technology is widely used in this new digital financial ecosystem [2], with over 100 million users worldwide and a total market capitalisation of cryptocurrencies exceeding the billion dollars.

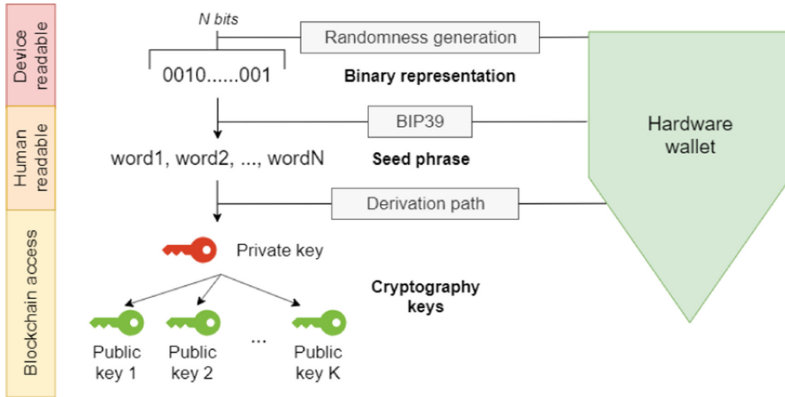
Thanks to cryptocurrencies, users hold a private cryptographic key to their financial sovereignty. To help users securely manage their private keys, electronic devices have emerged to enhance the safety of funds transactions: hardware wallets [3]. However, these purpose-built devices are often resource-constrained, with limited computational capabilities, memory capacity, and reduced entropy sources. Despite these limitations, they remain a popular choice for cryptocurrency holders.

This article aims to demonstrate that the resource-constrained nature of hardware wallets may lead to potential security attacks. In particular, three attacks against hardware wallets targeting their random wallet generation system and their limited memory are shown. Preliminary results have demonstrated the

feasibility of these attacks. The rest of the article is organised as follows: Sect. 2 presents some background notions, Sect. 3 describes the proposed security attacks, Sect. 4 provides some experimental results, and Sect. 5 closes the article.

## 2 Background

This section elaborates on the main concepts addressed in this article: hardware wallets, seed phrases and derivation paths. Figure 1 illustrates the relationship amongst these concepts.



**Fig. 1.** From bits to cryptographic keys in hardware wallets

### 2.1 Hardware Wallets

Hardware wallets have emerged as a reliable solution for safeguarding cryptocurrencies. These physical devices serve as secure offline storage, providing an impregnable fortress for digital assets against potential online threats. Unlike software wallets that run on computers or smartphones, hardware wallets are purpose-built devices designed solely for the secure management of cryptocurrencies. They operate in an isolated environment, commonly known as a “cold storage” setup, which ensures that the private keys are never exposed to the Internet. Hardware wallets offer two primary functionalities: (i) creating new wallets and (ii) restoring existing wallets. On the one hand, the hardware wallet needs to generate a new seed phrase with some randomness source for creating new wallets. On the other hand, hardware wallets allow users to restore a previously used wallet by introducing a seed phrase and, hence, regain access to the funds and transaction history. Some popular hardware wallet brands include Ledger, Trezor, and KeepKey [4,5].

Introducing randomness into a hardware wallet is essential to provide users with a truly random and secure cryptocurrency wallet. However, this implementation can pose several risks. For instance, hardware limitations can create challenges in generating sufficient levels of entropy required for cryptographic operations. When the randomness generation process is compromised or biased, it can lead to weak cryptographic keys or predictable seed phrases. This, in turn, exposes users' funds because adversaries might exploit patterns in the keys or phrases to gain unauthorised access.

## 2.2 Seed Phrases

Seed phrases were introduced to store the private keys in a secure and user-friendly fashion. A seed phrase (a mnemonic or recovery phrase) is a sequence of human-readable words generated deterministically from the private key. The relationship between the seed phrase and the private key is pivotal. When a wallet is created or restored, the seed phrase generates the private key and all the corresponding public addresses. A seed phrase may consist of 12, 15, 18, 21 or 24 words representing the underlying private key of 128, 160, 192, 224 and 256 bits, respectively.

Most hardware wallets work with seed phrases following the BIP39 (Bitcoin Improvement Proposal 39) standard, a widely adopted standard for generating seed phrases and their corresponding private keys. BIP39 aims to enhance the security and usability of mnemonic phrases by providing guidelines for generating seed phrases and converting them into deterministic seeds. Also, BIP39 defines a wordlist of 2048 unique words from which the words in the seed phrase are chosen.

## 2.3 Derivation Paths

In cryptocurrencies, derivation paths are hierarchical structures that generate and organise cryptographic keys to simplify key management and enhance compatibility. BIP32, BIP44, BIP49, BIP84, and BIP141 are some of the most common standards to generate derivation paths.

BIP32 introduces a hierarchical deterministic key generation method, allowing users to generate a master key from a random seed value. From this master key, a virtually infinite number of child keys can be derived deterministically, making it easier to manage multiple cryptocurrency addresses securely. Each derivation in the path is determined by a specific index (the values between the slashes), and the resulting keys can be used for receiving or sending funds. However, BIP32 does not define a specific structure for organising these keys, leading to the development of BIP44. BIP44 defines a multi-level derivation path that includes specific account indexes, making it easier for wallets to support multiple cryptocurrencies without mixing their keys. For example, a BIP44's derivation path for Bitcoin (BTC) might look like `m/44'/0'/0'`, where the first index (44') denotes the BIP44 standard, the second index (0') represents the BTC account,

and the third index ( $0'$ ) points to a specific address. From here, the last index might change (e.g.,  $1'$ ,  $2'$ , and so on) to obtain other cryptographic keys.

### 3 Attacks over Hardware Wallets

In this section, we describe three attacks against hardware wallets, namely (i) attacks based on low entropy, (ii) attacks based on uncommon derivation paths, and (iii) attacks based on public domain seeds. As a result of all these attacks, a list of public keys is obtained, which is then compared to online databases of public keys (e.g., Loyce.club) that have received transactions at some time. When matches are found, the wallet's private key is known (see Fig. 2).

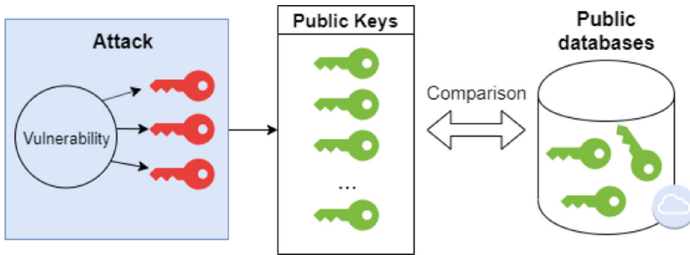


Fig. 2. Methodology of the security attacks conducted

#### 3.1 Attack #1: Low Entropy Attack

Low entropy refers to a state of limited randomness or unpredictability in data. In the context of cryptography, low entropy indicates that the information is relatively easy to guess or predict, thus being more vulnerable to attacks. This lack of randomness reduces the strength of seed phrases. Malfunctioning hardware wallets could potentially result in generating seed phrases with low entropy.

There are several mechanisms to generate strings with low entropy. One of these mechanisms consists in generating binary strings containing significantly more zeros than ones, or vice versa, e.g., “10001001000010000000010000001000” for a 32-bit representation (for the sake of simplicity). Errors during randomness exploitation in hardware wallets (e.g., repetitive character sequences or pattern-based sequences) lead to biased combinations of zeros and ones. A low entropy source provides a low entropy string generator. From these binary strings, the corresponding seed phrases are computed using BIP39. Then, public keys are obtained after deriving the seed phrases following BIP32 and BIP44, as hardware wallets do.

---

**Algorithm 1** Generation of low entropy strings

---

**Require:** $N =$  number of bits (typically,  $N = \{128, 160, 192, 224, 256\}$ ). $p =$  maximum number of ones ( $p < N$ ).**Ensure:**

wordList = list of all combinations of low entropy strings.

```

1: function STRINGGENERATOR( $N, p$ )
2:   wordList  $\leftarrow$  empty list
3:   for countOnes from 1 to  $N$  do
4:     for i from 0 to  $p$  do
5:       binary  $\leftarrow$  0;
6:       for j from 0 to countOnes - 1 do
7:         position  $\leftarrow$  (i + j) %  $N$ 
8:         binary  $\leftarrow$  binary OR (1 << position)
9:         wordList.add(binary)
10:      end for
11:    end for
12:  end for
13:  return wordList
14: end function

```

---

### 3.2 Attack #2: Uncommon Derivation Path Attack

Derivation paths are commonly generated using the BIP standards described in Sect. 2.3. However, derivation paths could be generated without following any standard, i.e., customising the indexes of the derivation path.

This attack builds upon the strings with the lowest entropy obtained from the previous attack. The lower the strings' entropy, the more possibilities that might correspond to someone's public addresses. So, after generating the corresponding seed phrases from these strings, it is worth applying different kinds of derivations (i.e., considering uncommon derivation paths too).

In particular, two approaches have been followed to generate uncommon derivation paths. The first approach creates custom derivation paths by building all the combinations using the most frequent index values. As observed, these values are 0, 0', 1, 1', e.g.,  $m/0/1/0$ ,  $m/0'/1/0'$  and  $m/0'/1/0/0$ . This way, a new set of public keys is obtained. The second approach analyses how derivation paths are implemented in popular programming libraries, such as Python's *bip32utils*. With this approach, their funds could be compromised if users have started from strings with very low entropy for their key generation when using these libraries.

### 3.3 Attack #3: Public Domain Seed Phrases Attack

This attack leverages the functionality of hardware wallets to reset an existing wallet by inputting a previously generated seed phrase. Also, this functionality can be used by users to create a new wallet by deciding their seed phrase. Unfortunately, hardware wallets do not have any mechanism to alert users that their seed phrases are weak, e.g., publicly available on the Internet. A strategy to discover weak seed phrases involves seeking the least common words from the BIP39's wordlist on the Internet, e.g., using tools like Pastebin or PrivateBin. If seed phrases are found, public keys could be derived from them.

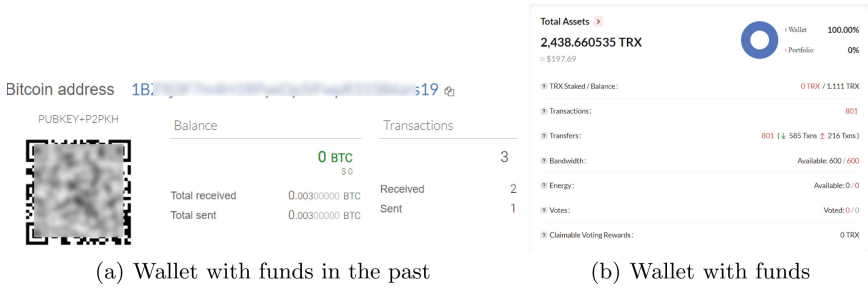


Fig. 3. Wallets discovered

## 4 Initial Preliminary Results

Some preliminary tests have been conducted to prove the attacks' feasibility. Access has been gained to several wallets with transactions involving different cryptocurrencies (see Fig. 3). Also, although numerous empty wallets have been found, they had funds for a long time. If they receive funds from now on, they could be compromised. The most significant findings are listed as follows:

- By using Attack #1, a Ripple (XRP) wallet with 490.64932 XRP (\$337 worth at this time) was discovered. This wallet is active with several weekly transactions.
- By using Attack #2, by exploiting the *bip32utils* library, some wallets with no current funds were found. However, one had 0.003 BTC (\$87 worth) for three months.
- The highest success rate was achieved applying Attack #3. A Tron (TRX) wallet with 197 USDT (\$197 worth) was discovered. This wallet is active, with many transactions every week.

## 5 Conclusions

Hardware wallets are essential to safeguard cryptocurrencies securely and reliably. However, the resource-constrained nature of these devices opens the door to numerous vulnerabilities. In this article, we have described three attacks against these devices: attacks against their low entropy, attacks against uncommon derivation paths, and attacks against public domain seed phrases. Some wallets with recent activity, and even with funds, have been discovered. Further work will concentrate on analysing the electronic components of hardware wallets during the generation of the private key (e.g., side-channel attacks) and evaluating the functioning of these components under stressful conditions, such as extreme temperatures.

**Acknowledgements.** This project has been partially funded by AGAUR research group 2021SGR-00111: “ASCLEPIUS”.

## References

1. Fang F et al (2022) Cryptocurrency trading: a comprehensive survey. *Financ Innov* 8(1):1–59
2. Casino F et al (2019) A systematic literature review of blockchain-based applications: current status, classification and open issues. *Telemat Inform* 36:55–81
3. Arapinis M et al (2019) A formal treatment of hardware wallets. In: 23rd international conferences financial cryptography and data security, pp 426–445. Springer
4. Dabrowski A et al (2021) Better keep cash in your boots—hardware wallets are the new single point of failure. In: *Proceedings of ACM CCS workshop on decentralized finance and security*, pp 1–8
5. Almutairi E et al (2019) Usability and security analysis of the keepkey wallet. In: *IEEE international conferences blockchain and cryptocurrency*. IEEE, pp 149–153