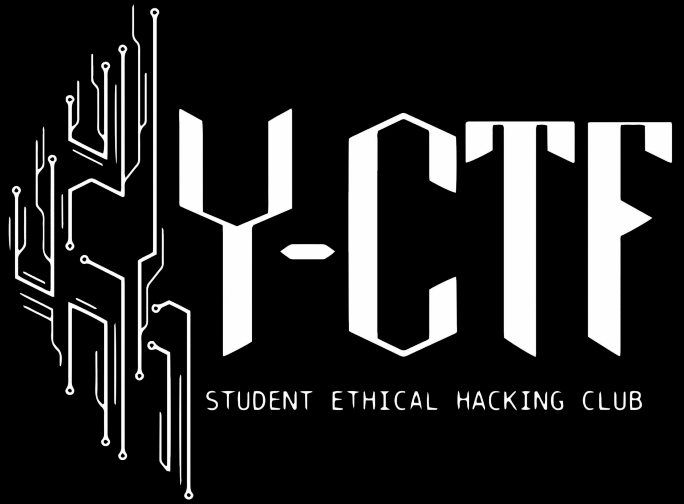
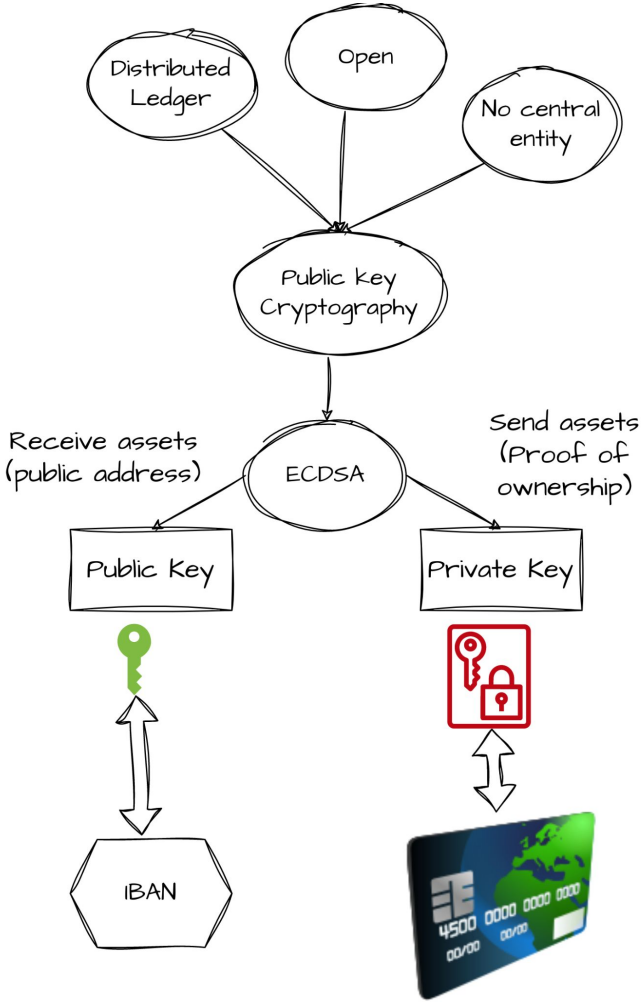
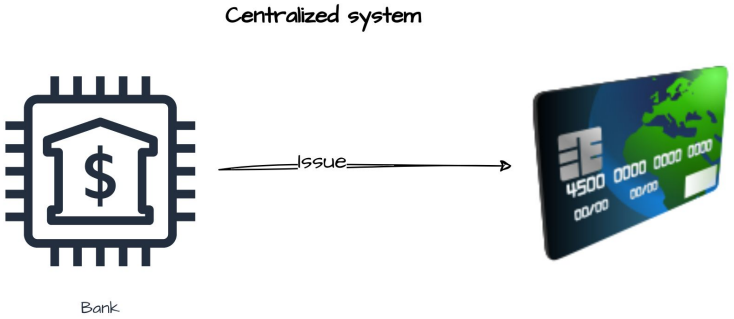
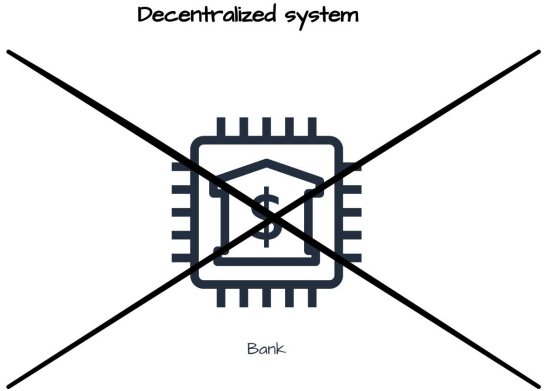
Crypto Wallet
What Could Go Wrong ?



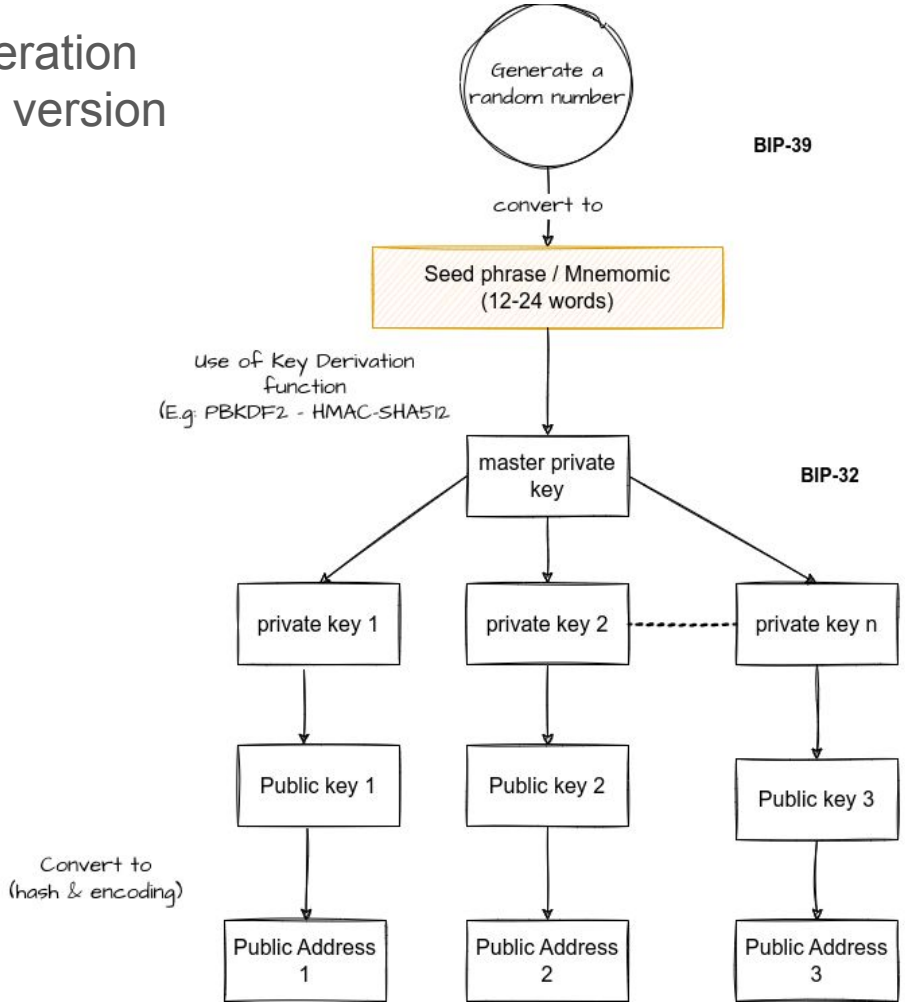
Blockchain - Introduction





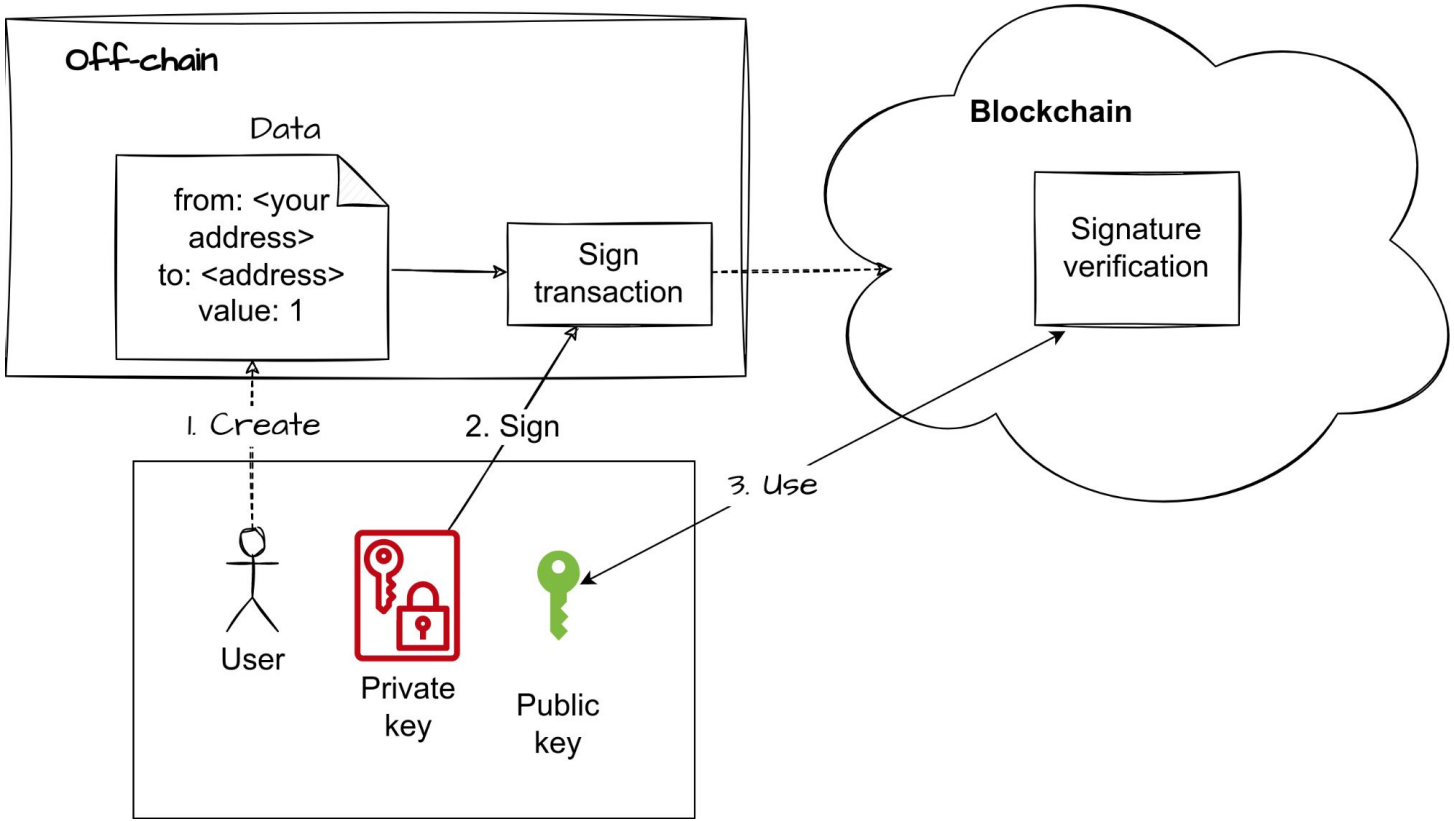


Keys generation
Simplified version

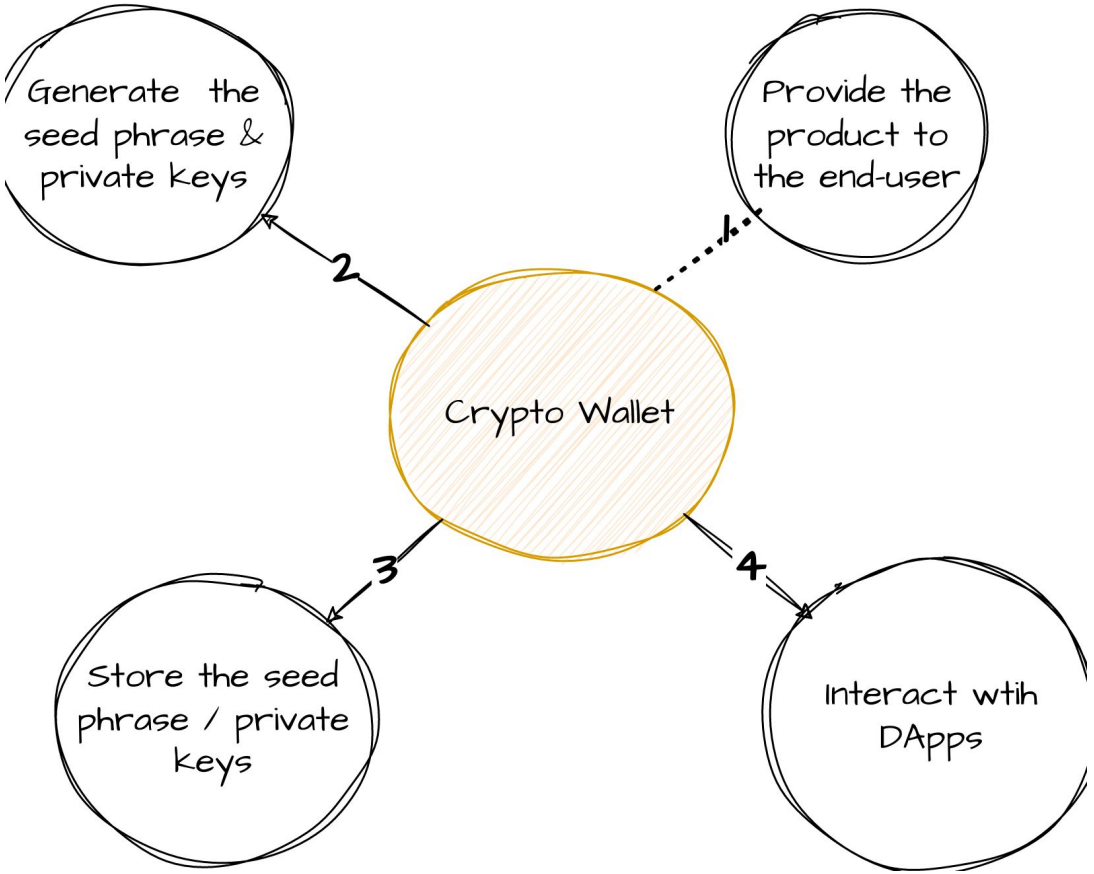


1.	private	13.	will
2.	digital	14.	prevent
3.	coin	15.	animal
4.	seed	16.	weasel
5.	key	17.	brain
6.	have	18.	person
7.	very	19.	like
8.	long	20.	you
9.	secret	21.	obtain
10.	pass	22.	any
11.	phrase	23.	large
12.	that	24.	wealth

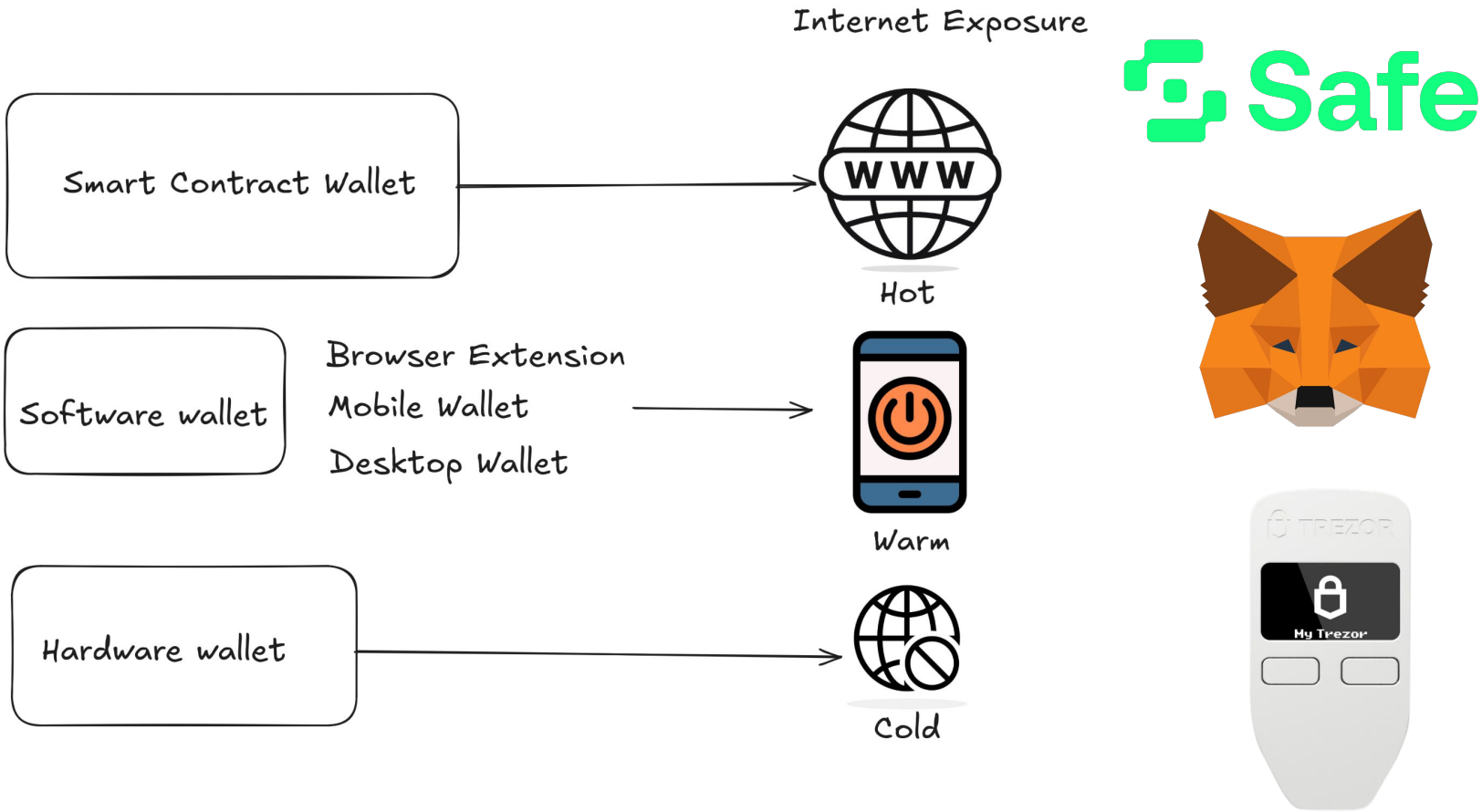
Transaction - Basic workflow

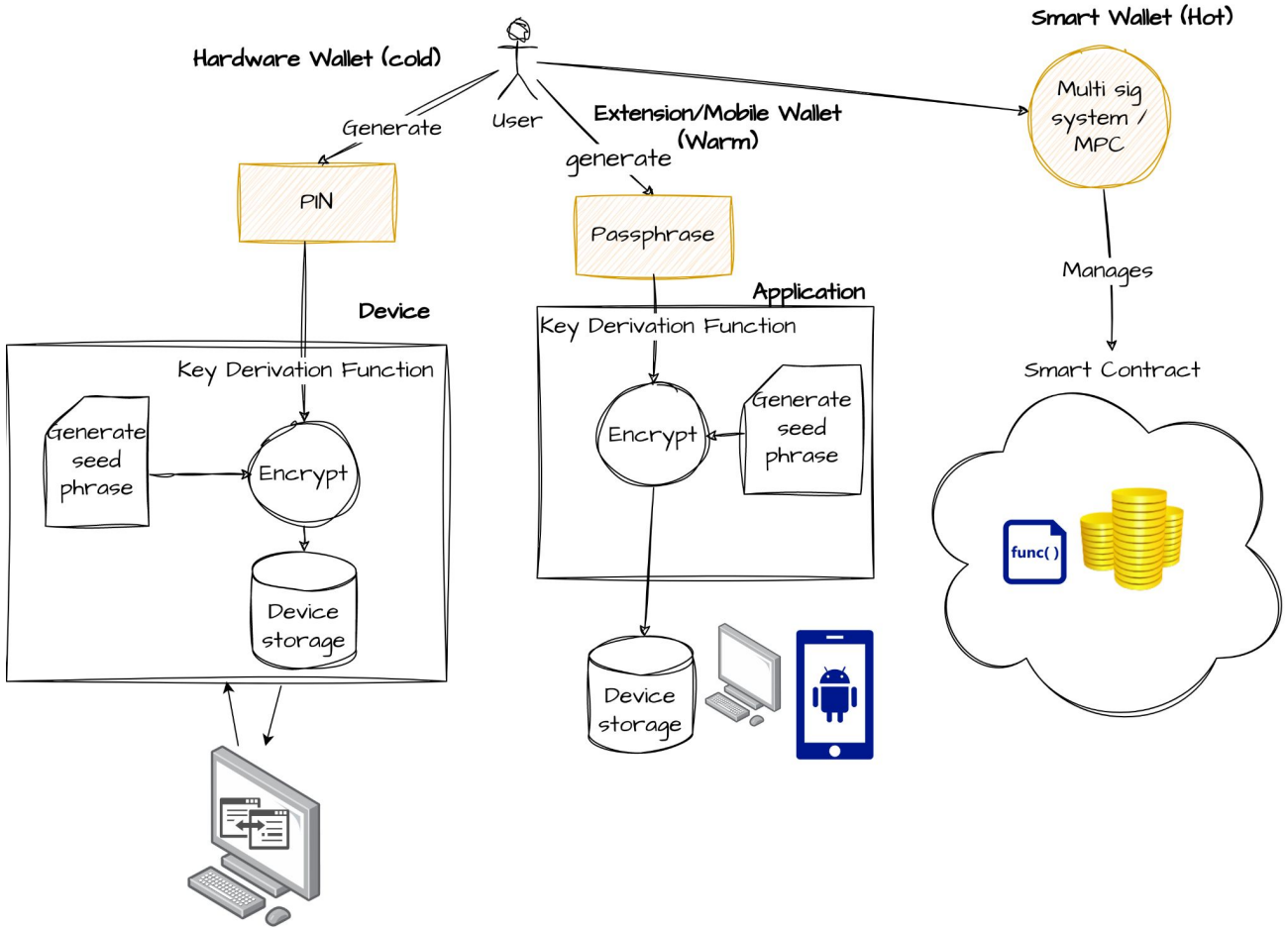


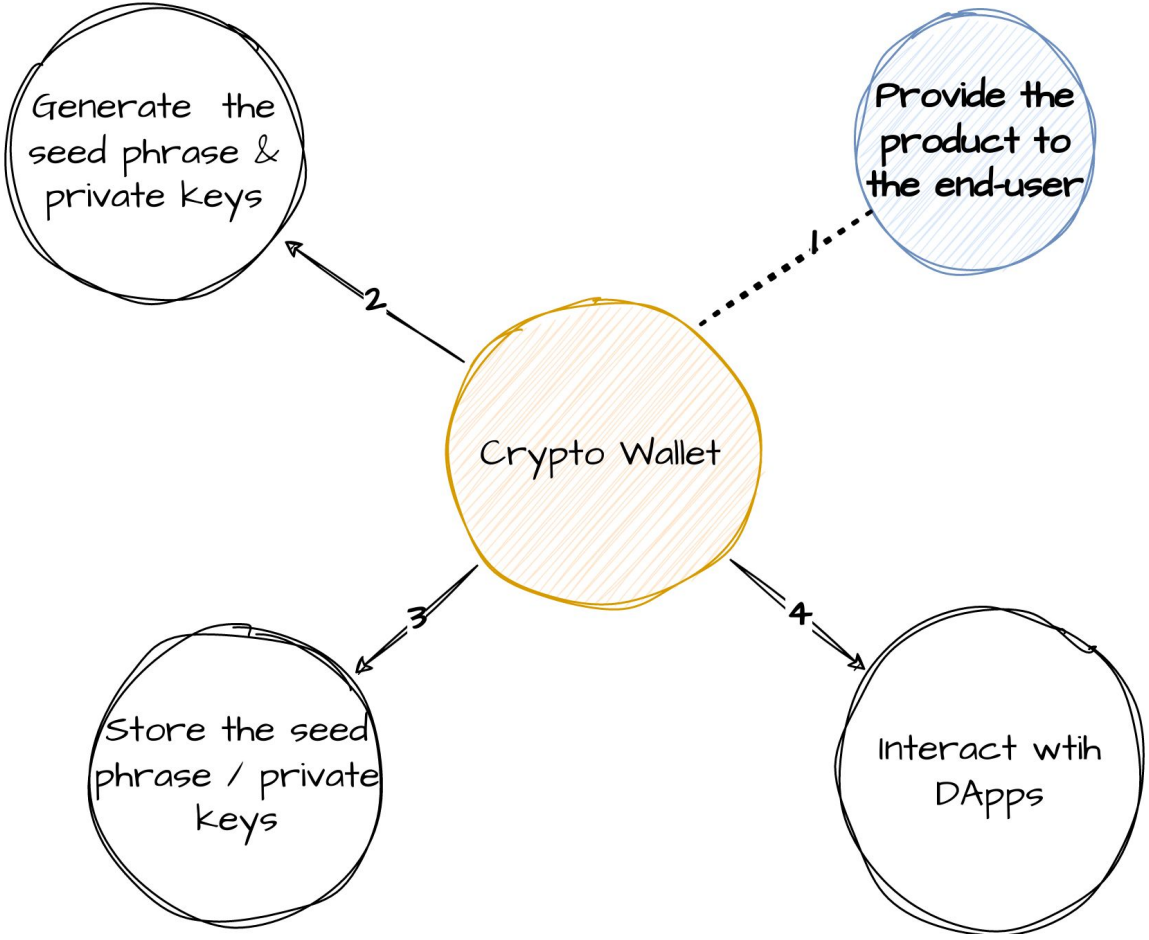
Crypto Wallet Task

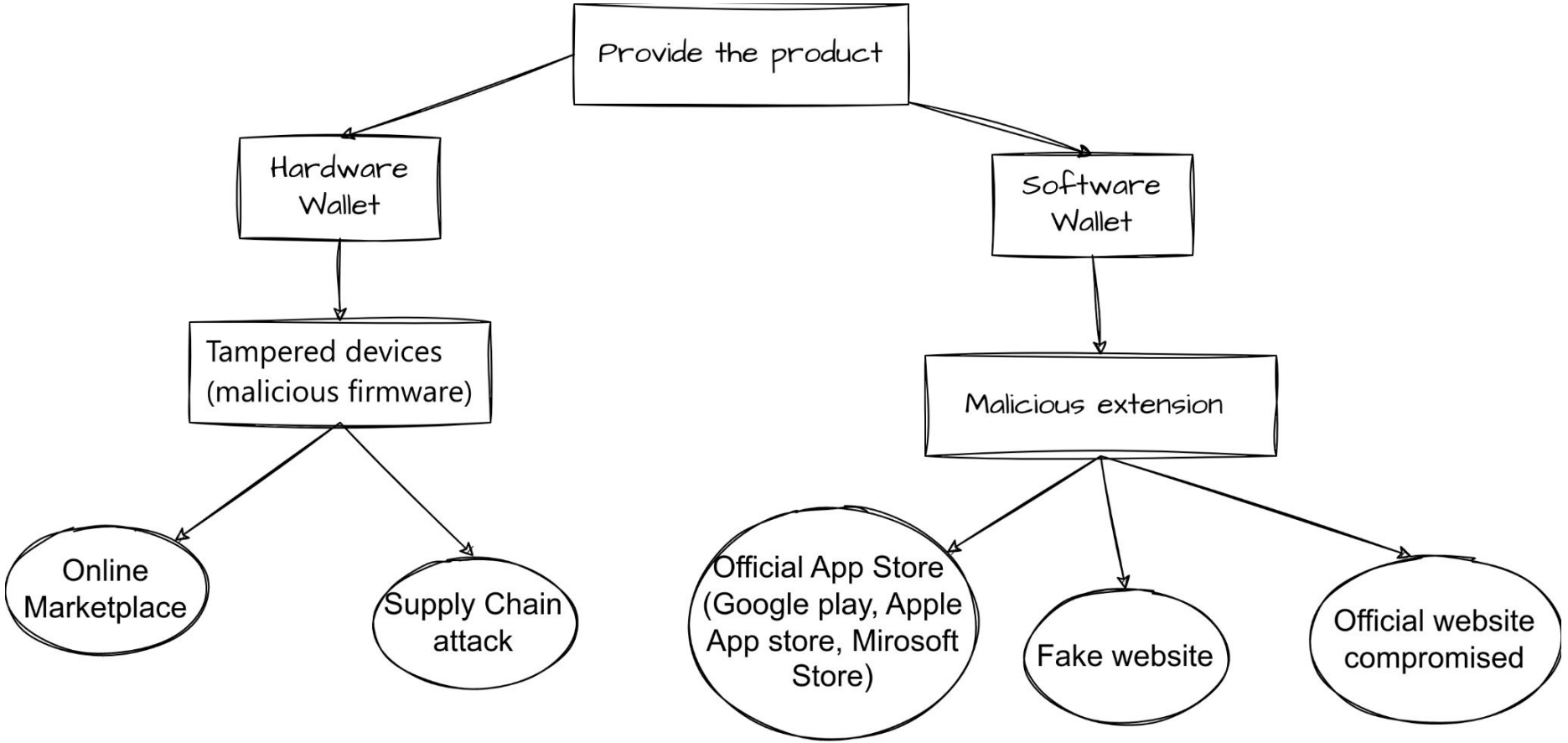


Type of Wallets



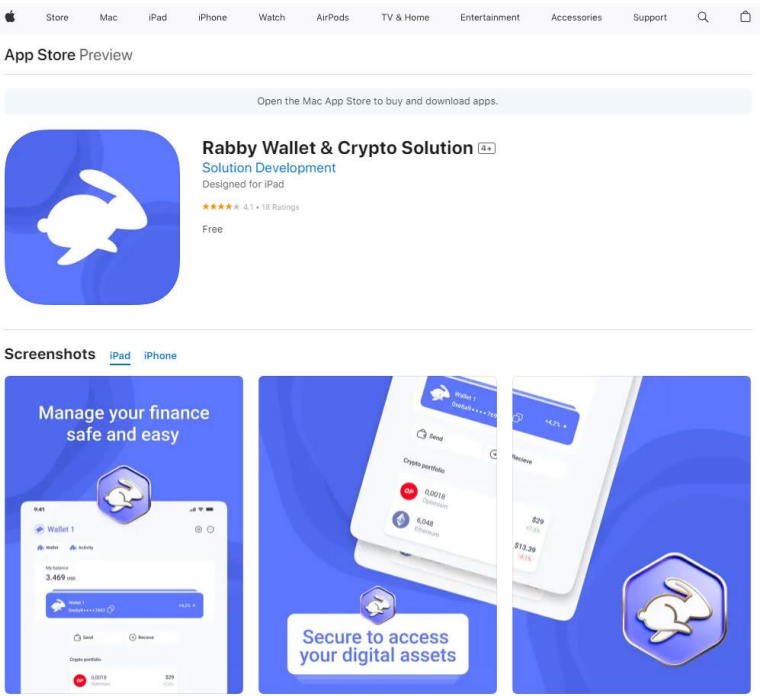

Hardware Wallet (cold)
User
Smart Wallet (Hot)
Generate
Extension/Mobile Wallet
(Warm)
generate
Multi sig
system /
MPC
PIN
Passphrase
Manages
Device
Application
Key Derivation Function
Key Derivation Function
Smart Contract
Generate
seed
phrase
Encrypt
Generate
seed
phrase
Encrypt
func( )
Device
storage
Device
storage


Generate the seed phrase & private keys
Provide the product to the end-user
1
2
Crypto Wallet
3
4
Store the seed phrase / private keys
Interact wtih DApps


Provide the product
Hardware Wallet
Software Wallet
Tampered devices (malicious firmware)
Malicious extension
Online Marketplace
Supply Chain attack
Official App Store (Google play, Apple App store, Mirosoft Store)
Fake website
Official website compromised


Software wallet

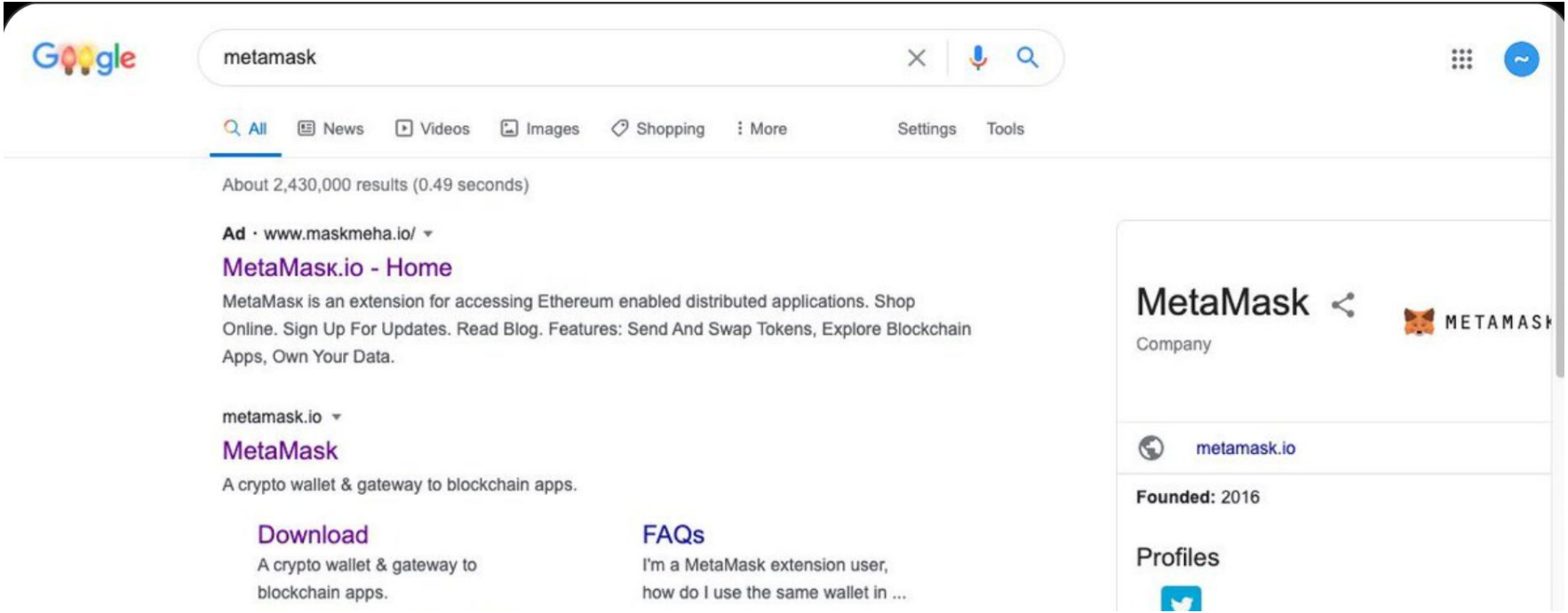
Fake extension on official store (2024)



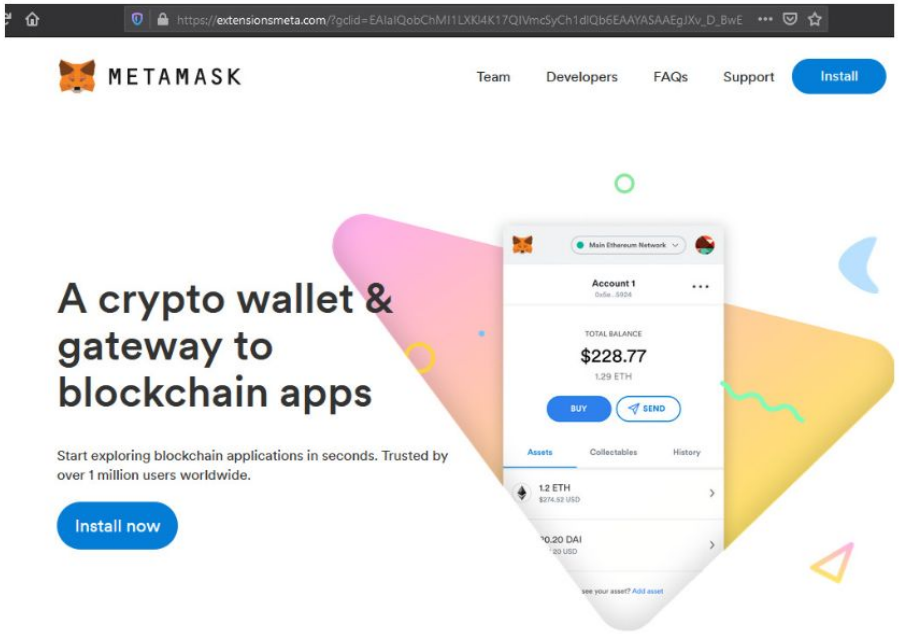
- Rabby Wallet is a browser extension (Warm wallet)
- At the time of the attack, an official version for Apple was under review by Apple but not published
- Attackers published the malicious version on the Apple Store
- Around $40,000 worth of crypto stolen
- Another similar attack generated $600,000 in stolen crypto (Fake Ledger Live App, 2023)

[cointelegraph - Fake Rabby wallet wreaks havoc after listing on Apple app store]

Malicious Metamask Ads (2020)

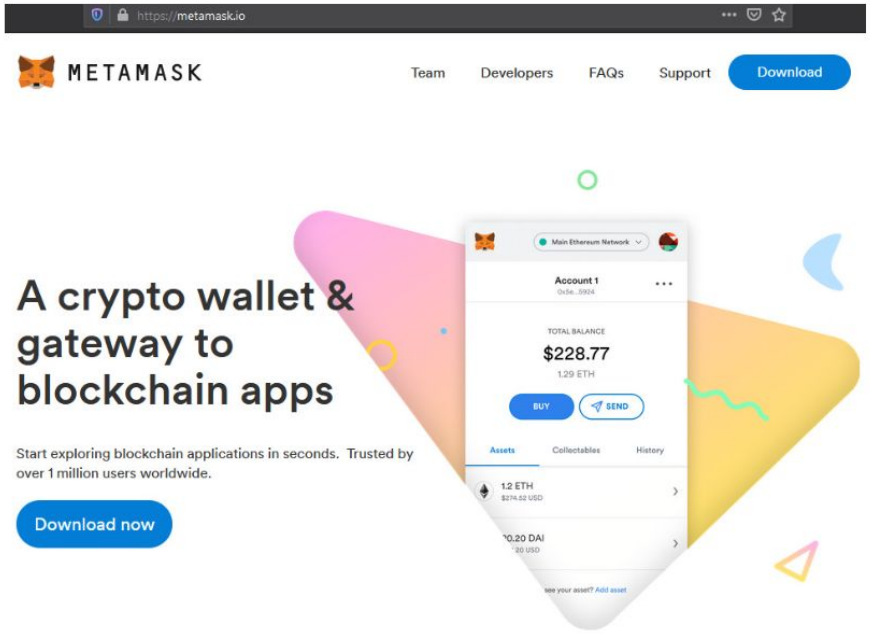


[bleepingcomputer.com - MetaMask phishing steals cryptocurrency wallets via Google ads]
[Metamask - Link]



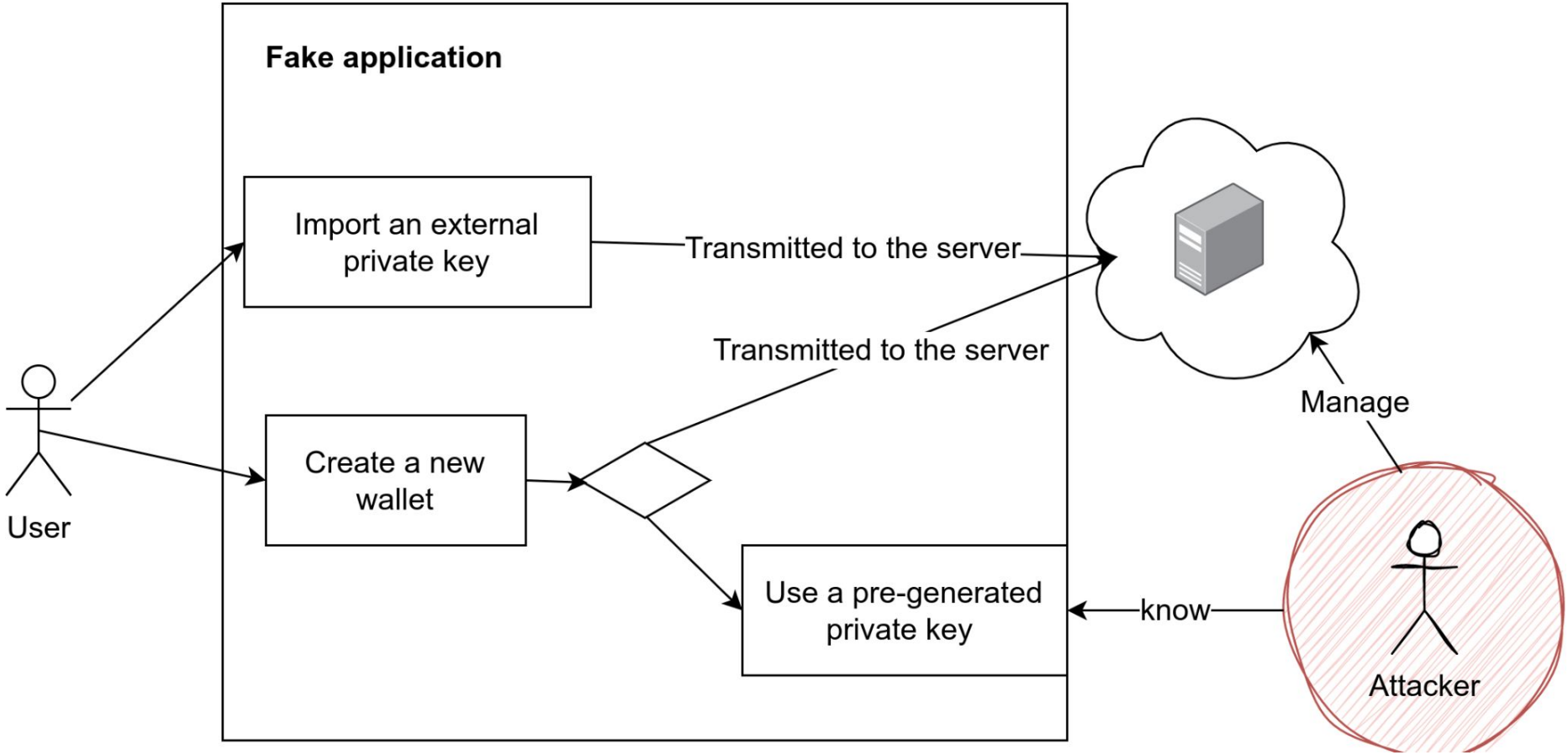
LEARN MORE

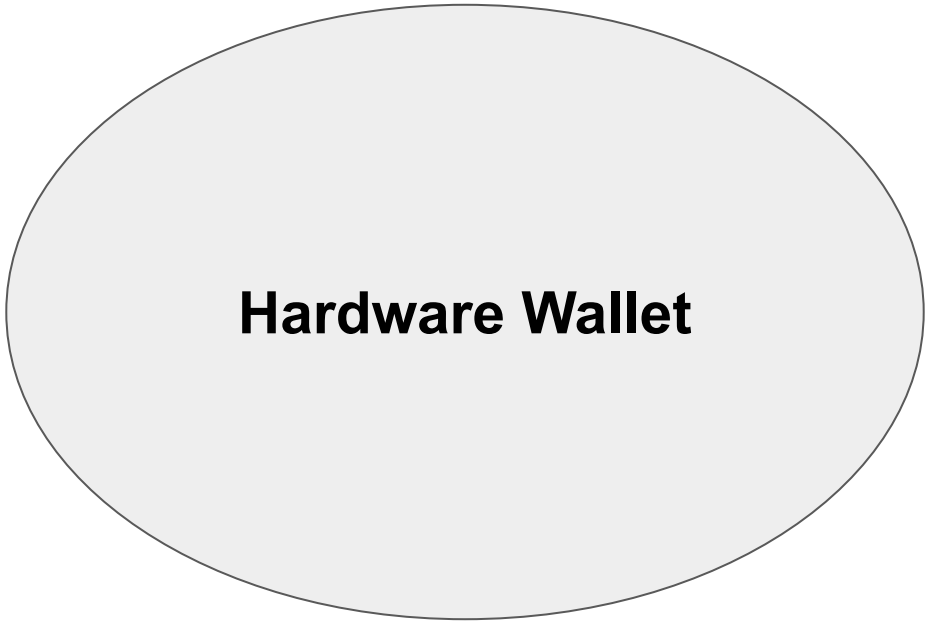
Fake MetaMask site



LEARN MORE

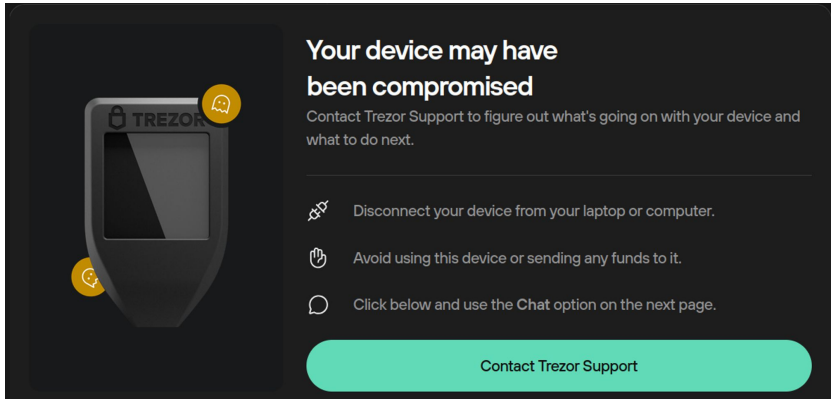
Legitimate MetaMask site


Fake application
Import an external private key
Transmitted to the server
Transmitted to the server
Create a new wallet
Use a pre-generated private key
User
Manage
know
Attacker


Hardware Wallet

Hardware Wallet - Trezor Measure

- Seal on the device or the package.
- A legitimate device will always arrive without firmware installed
- The bootloader verifies that the firmware you install has been signed by SatoshiLabs (= secure boot).
- “New”: individual chips are now glued onto the board



Fake Trezor Wallet (2022)

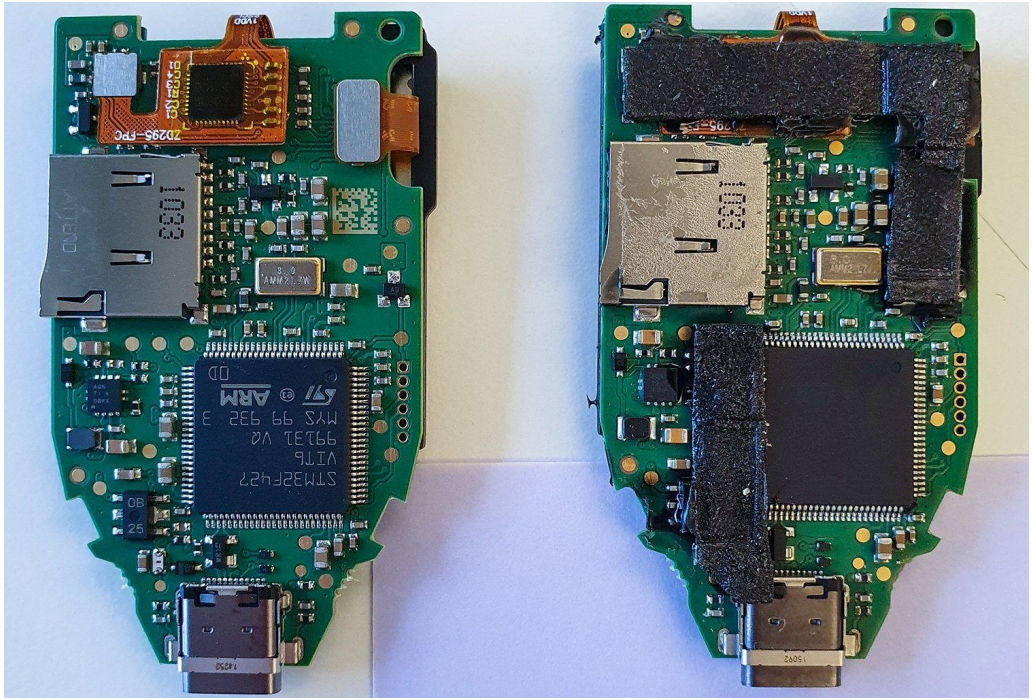
Fake Trezor sold on the Russia Market through a popular website

Modification performed:

1. They install a malicious firmware in the device
2. They removed the bootloader-checks to avoid detection at startup.
3. They replace the randomly generated seed phrase with a pre-generated seed phrases saved in the malicious firmware.

Kaspersky.com - Case study: fake hardware cryptowallet

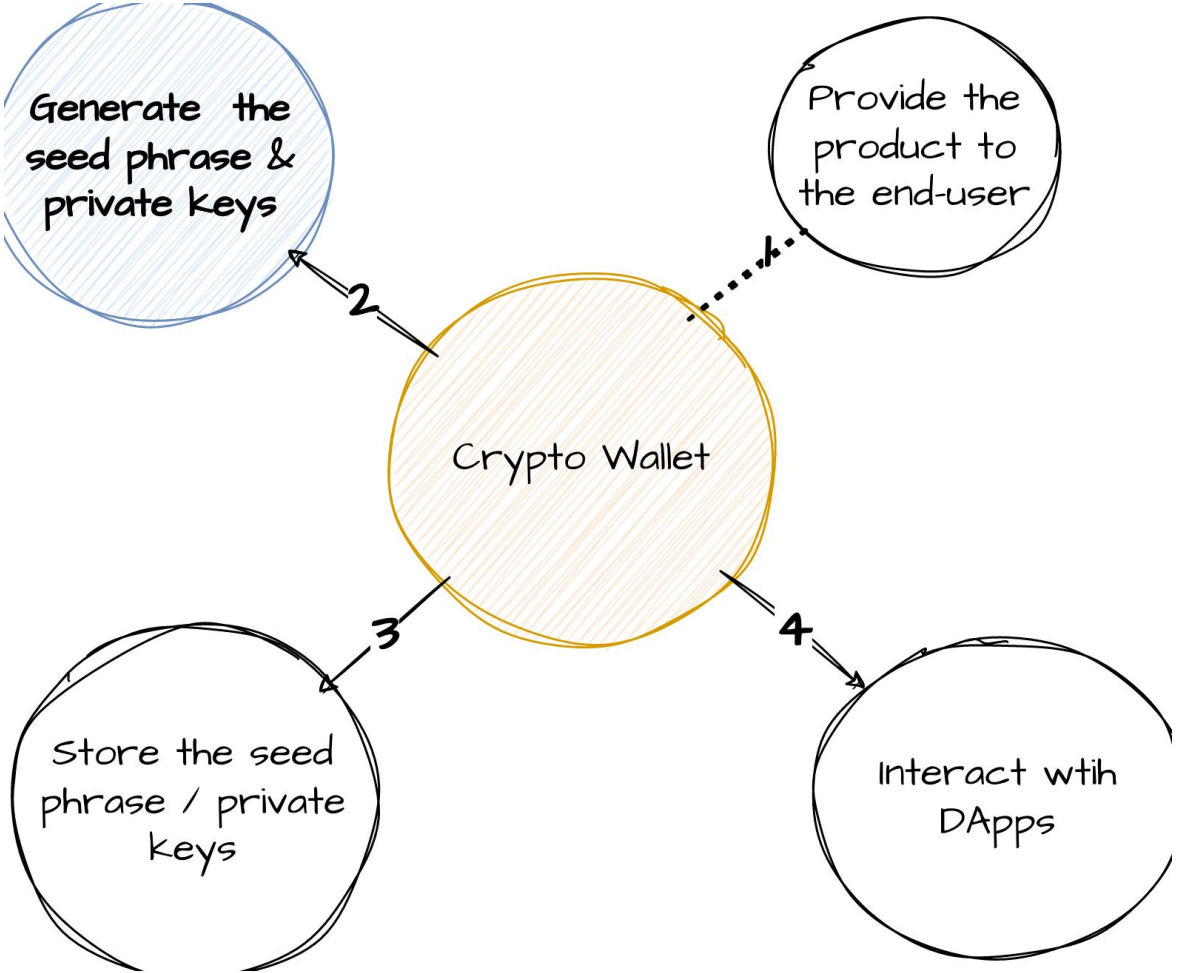
2022 - Fake Trezor on Russia Market

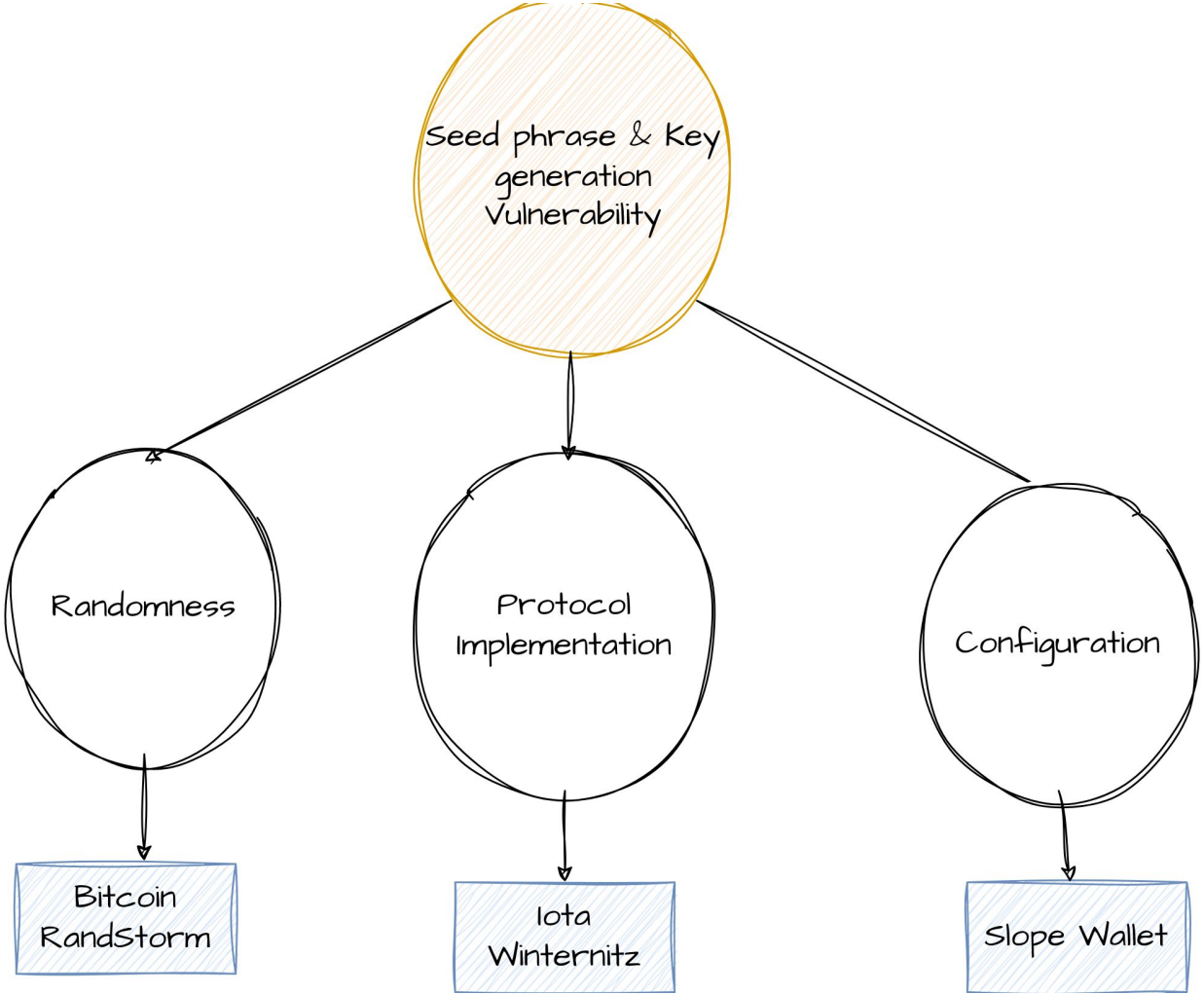


Official Device

Fake Device

[Kaspersky.com - Case study: fake hardware cryptowallet]


Generate the seed phrase & private keys
Provide the product to the end-user
1
2
Crypto Wallet
3
4
Store the seed phrase / private keys
Interact wtih DApps


Seed phrase & Key generation Vulnerability
Randomness
Protocol Implementation
Configuration
Bitcoin RandStorm
Iota Winternitz
Slope Wallet

Bitcoin JS - Bitcoin RandStorm - 2011<->2015

- BitcoinJS was a popular package used for browser based crypto wallets.
- The vulnerability concerns wallets generated by BitcoinJS (and derivative projects) between 2011 - 2015.
- Discovered a first time in 2018, and again in 2022 by Unciphered, a cryptocurrency recovery company
- No funds stolen
- Reference:
  - unciphered.com - Randstorm: You Can’t Patch a House of Cards
  - unciphered.com - Randstorm
  - Kaspersky - Randstorm: vulnerable crypto wallets from the 2010s

Modern Wallet: Web Crypto API (getRandomValues)

IOTA Winternitz - 2017

- IOTA used Winternitz as the main signature scheme for their ledger
- Winternitz consists of computing each element in a chain by hashing the previous element.

$$X_i = HASH(X_i - 1)$$

- They design their own hash function *Curl*, based on the Sponge construction.
- They also used it to hash the message

Vulnerability

- *Curl* was vulnerable to collision attack through differential cryptanalysis.
- A technique mainly used against block cipher (known attack against FEAL) popularized in the 1980s.
- Using this collision attack, it is possible to forge valid signature.

Condition: Chosen plaintext attack

For example:

- An attacker creates two payments
  - A benign payment and a malicious payment
  - A signature on the benign payment by the target is also a valid signature on the malicious payment.
  - Mainly an attack on multi-signature payments: one user must produce a payment for another user to sign.

Reaction from the IOTA foundation

- According to the foundation, the vulnerability was difficult to exploit in practice
- Notably, due to presence of the “coordinator”, a centralized black box component managed by the IOTA foundation used to validate transactions
- Nevertheless, they switch for another hash function *Kerl* based on Keccak (SHA-3)
- Now, IOTA don’t use any longer Winternitz, but only EdDSA

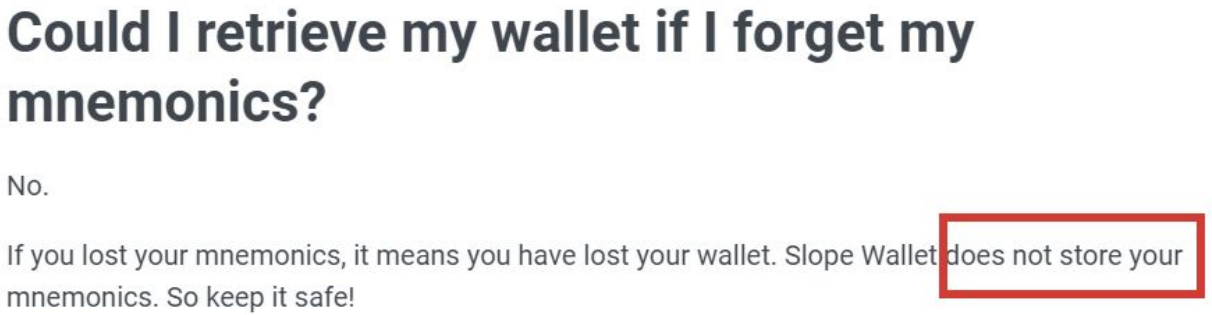
IOTA - Upgrades & Updates

github.com - vuln-iota.md, eprint - Cryptanalysis of Curl-P and Other Attacks on the IOTA Cryptocurrency

Slope Wallet - 2022

Bad Configuration

- Blockchain: Solana
- Victims: 9,231 wallets for about $4 millions stolen.



Could I retrieve my wallet if I forget my mnemonics?

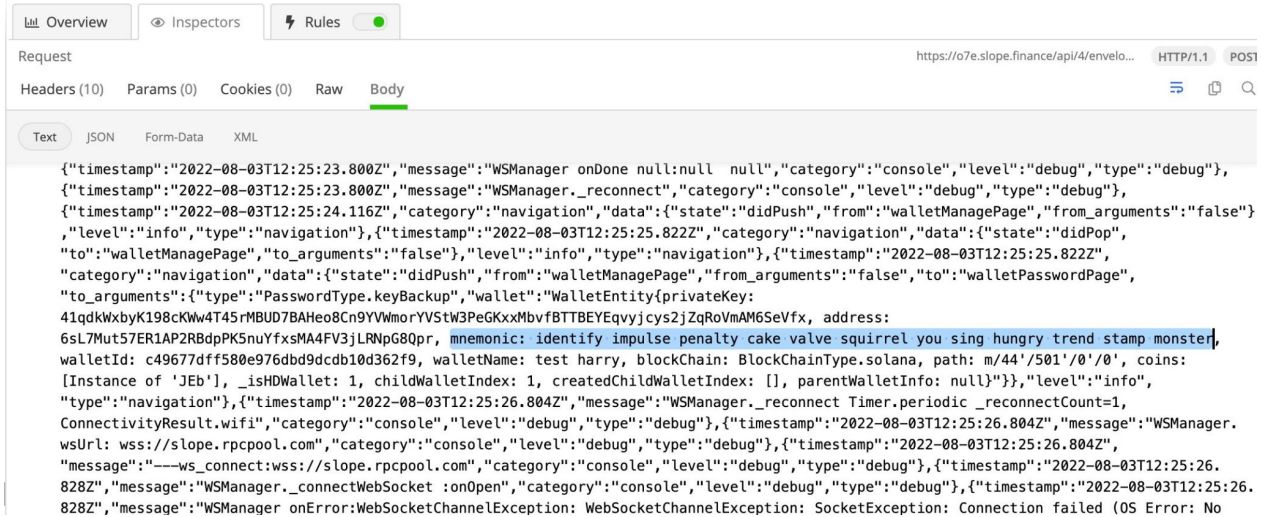
No.

If you lost your mnemonics, it means you have lost your wallet. Slope Wallet does not store your mnemonics. So keep it safe!

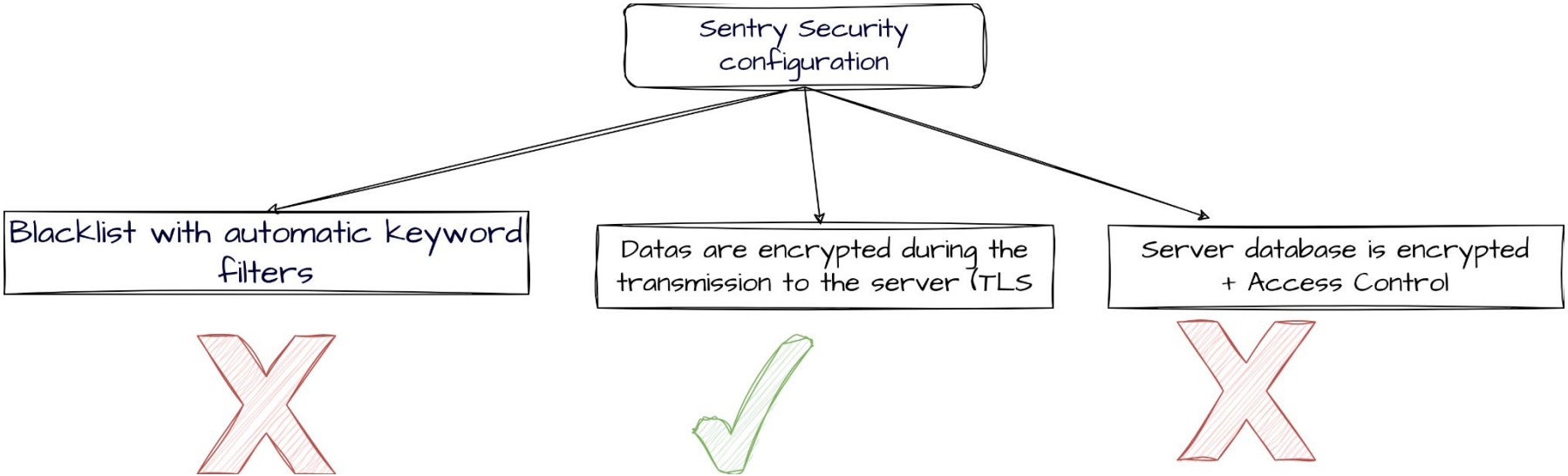

Self custodial wallet
Seed phrase is stored on the device
Use Sentry to log events
Log the seed phrase
imgflip.com

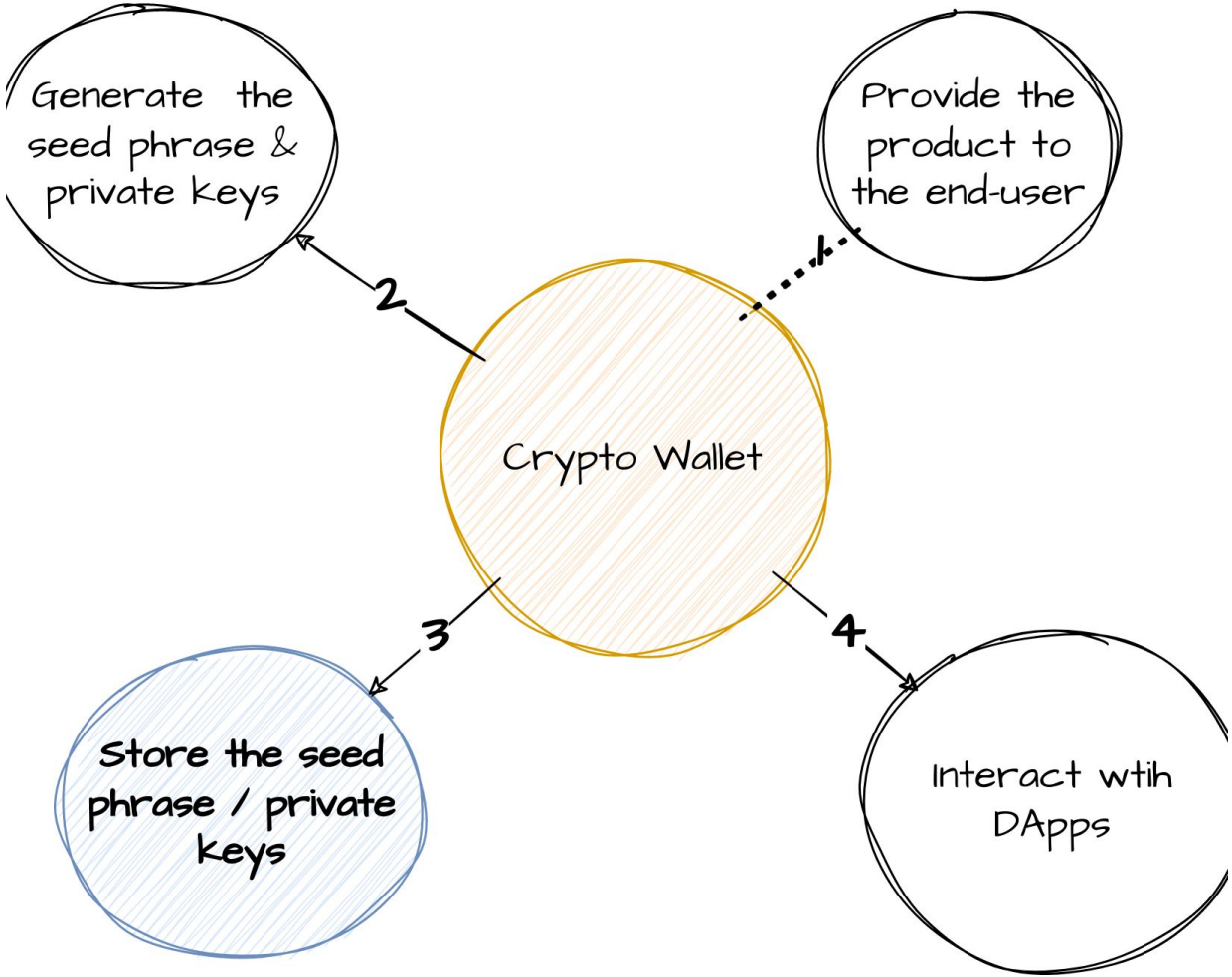
Slope Hack - Summary

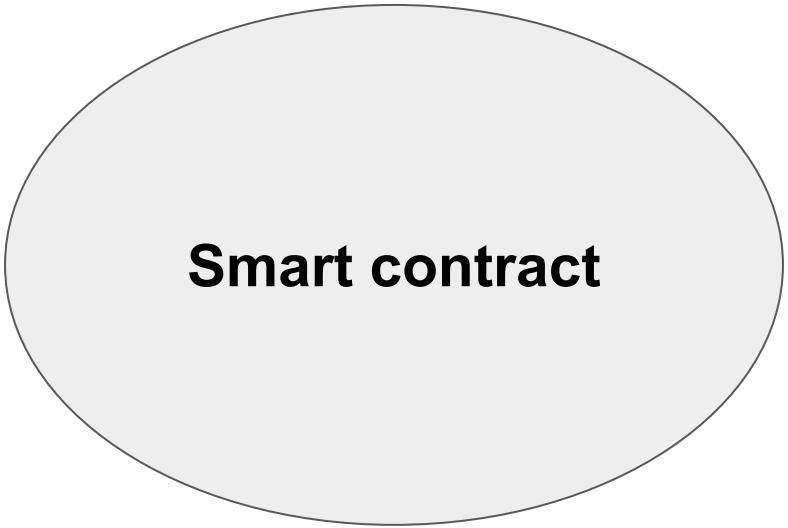
- Slope's mobile app sends off the seed phrase via TLS to their centralized Sentry server.
- The seed phrases are then stored in plaintext.
- Anybody with access to Sentry could access user seed phrase.



[OtterSec - Tweet]
[Tristan0x - Tweet]
[Solana - Slope Wallet Accident Update]


Sentry Security configuration
Blacklist with automatic keyword filters
Datas are encrypted during the transmission to the server (TLS
Server database is encrypted + Access Control


Generate the seed phrase & private keys
Provide the product to the end-user
1
2
Crypto Wallet
3
4
Store the seed phrase / private keys
Interact wtih DApps


Smart contract

anyone can kill your contract #6995

Closed **ghost** opened this issue on Nov 6, 2017 · 17 comments

ghost commented on Nov 6, 2017 • edited by ghost

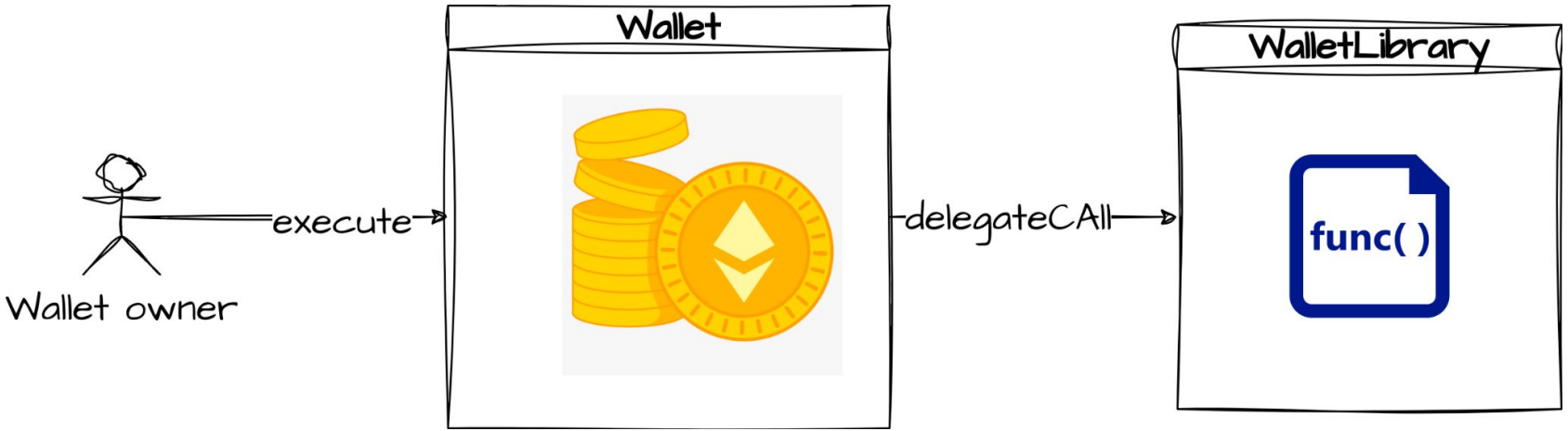
I accidentally killed it.

https://etherscan.io/address/0x863df6bfa4469f3ead0be8f9f2aae51c91a907b4

👍 75 👎 4 😄 123 🎉 64 😕 24 ❤️ 52 🚀 3 👀 4

Parity Bug - 2017

- Until this year (2024), Ethereum allowed to remove code from the blockchain with a specific instruction (self destruct)
- Parity wallet used a library deployed on Ethereum for its wallet



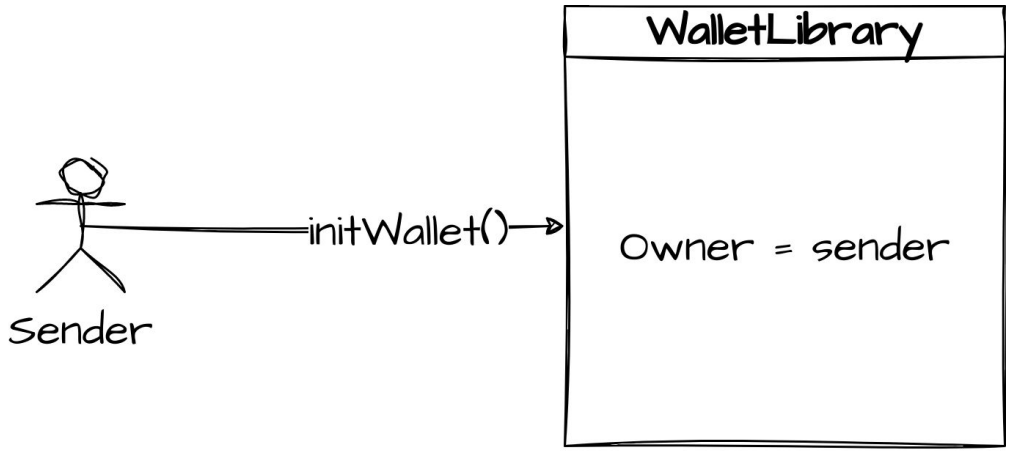
Details

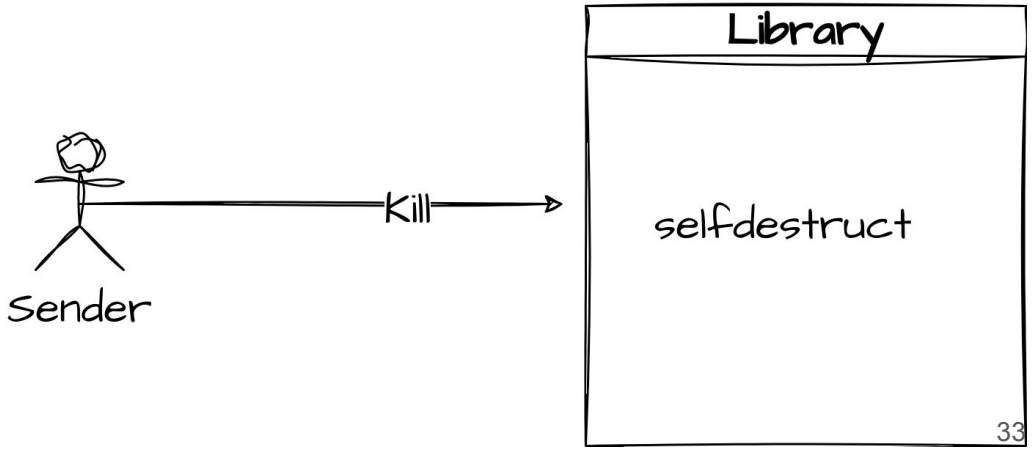
Context

- This library contained a *self destruct* operation (function *kill*)
- This library did not have an admin configured.

Step

- Someone managed to take ownership of the library
- Then triggered the *selfdestruct* operation through the function *kill*

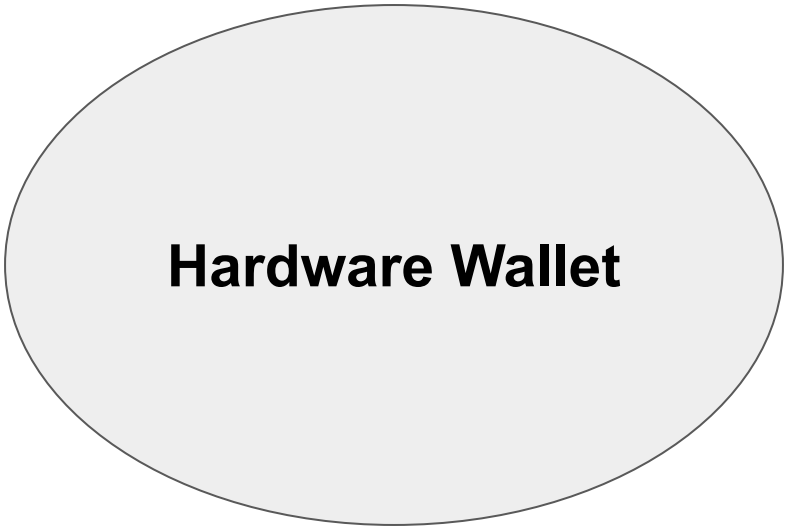




Consequence

- Freeze ~500 multi-signature wallets
- Irreversibly freezing ~$150M.
- Hard fork required to restore the contract and/or return funds.

researchgate - Smart Contracts Vulnerabilities: A Call for Blockchain Software Engineering?


Hardware Wallet

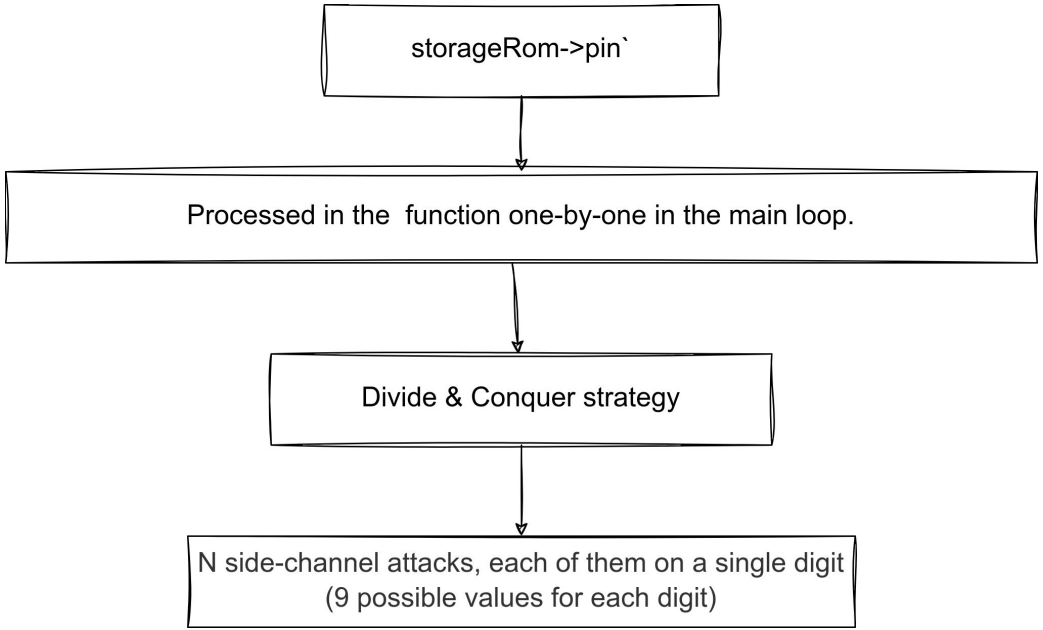
Trezor - 2018-2019

- Old Trezor model one
- 2018-2019
- Identified by Ledger & Wallet.fail


```c
/* Check whether pin matches storage. The pin must be
* a null-terminated string with at most 9 characters.
*/
bool storage_containsPin(const char *presented_pin)
{
    /* The execution time of the following code only depends on the
    * (public) input. This avoids timing attacks.
    */
    char diff = 0;
    uint32_t i = 0;
    while (presented_pin[i]) {
        diff |= storageRom->pin[i] - presented_pin[i];
        i++;
    }
    diff |= storageRom->pin[i];
    return diff == 0;
}
```

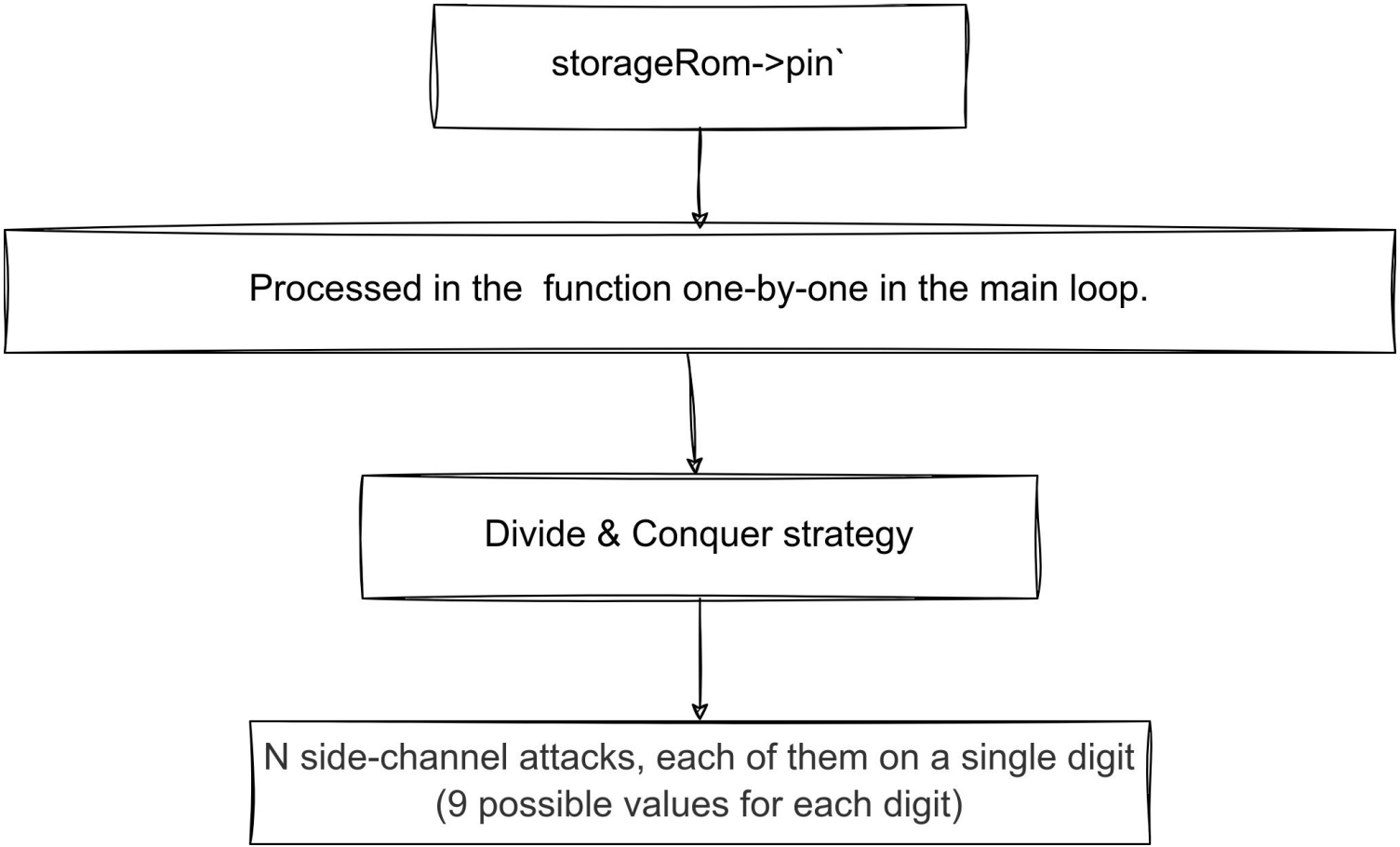
Old Trezor one - from Breaking Trezor One with Side Channel Attacks

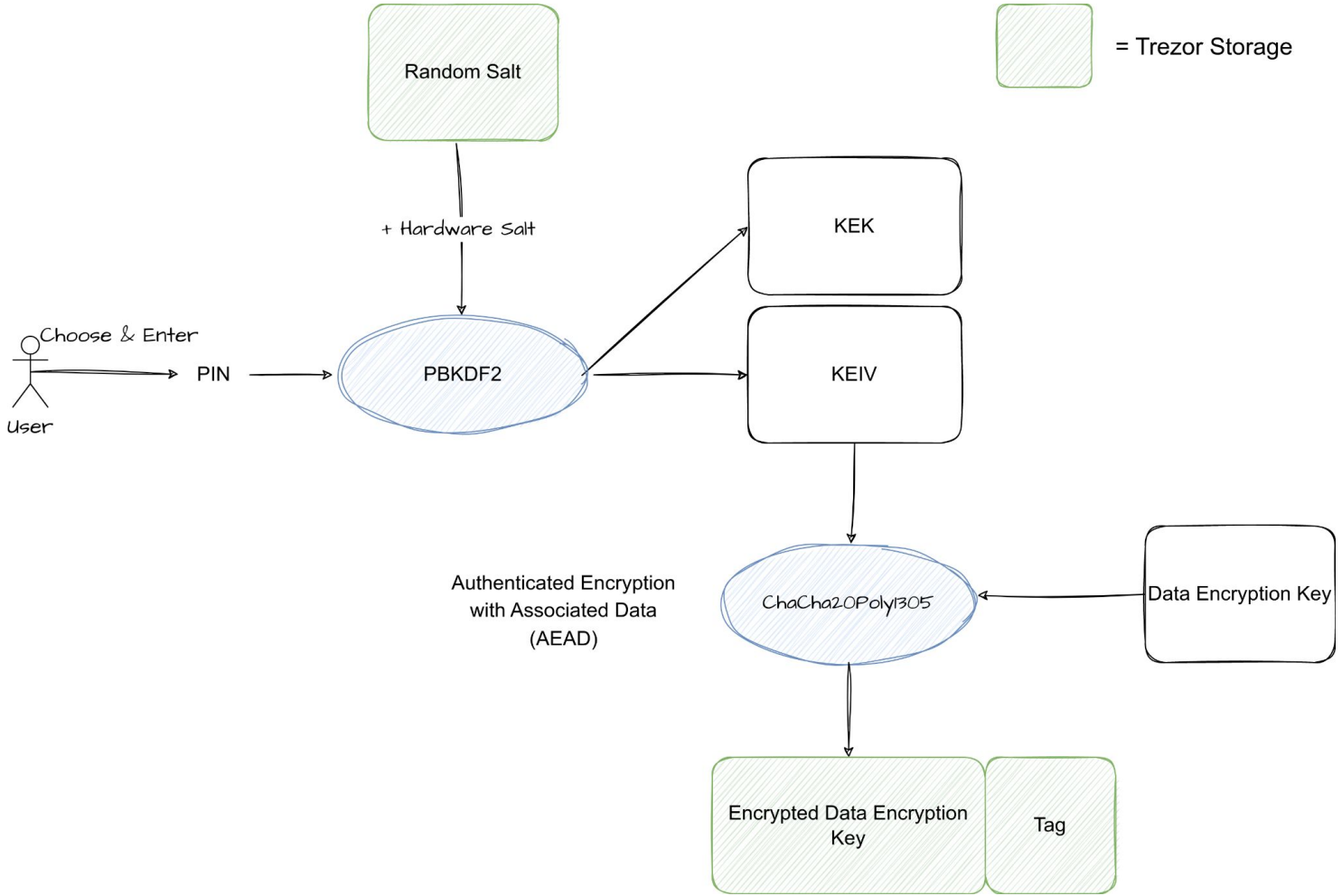
```
while (presented_pin[i]) {
    diff |= storageRom->pin[i] - presented_pin[i];
    i++;
}
```



The subtraction operation:
`storageRom->pin[i] − presented_pin[i]`

- Contains the secret PIN value and the user input value
- side-channel attack to induce differentiability.
  - They measure the power consumption of the device
  - They use Machine Learning to determine the most likely candidate for *storageRom->pin.*

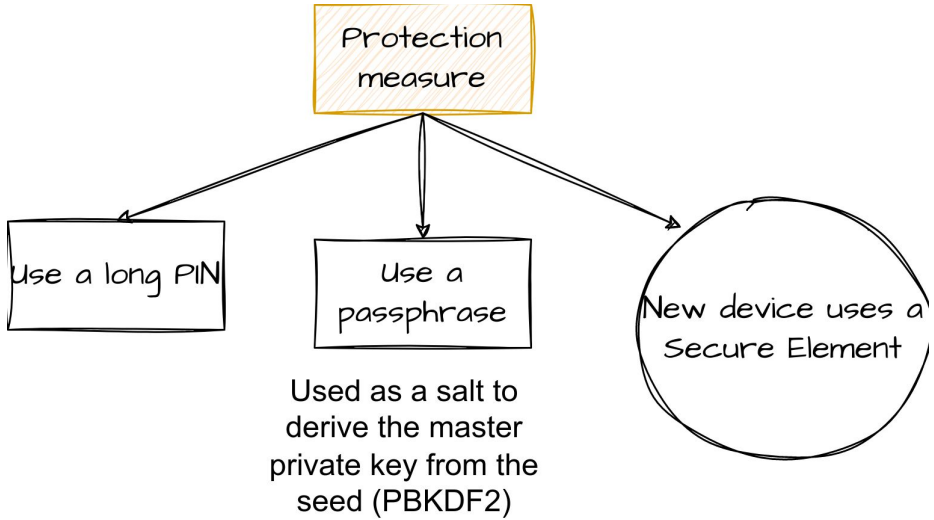

storageRom->pin`
Processed in the function one-by-one in the main loop.
Divide & Conquer strategy
N side-channel attacks, each of them on a single digit
(9 possible values for each digit)

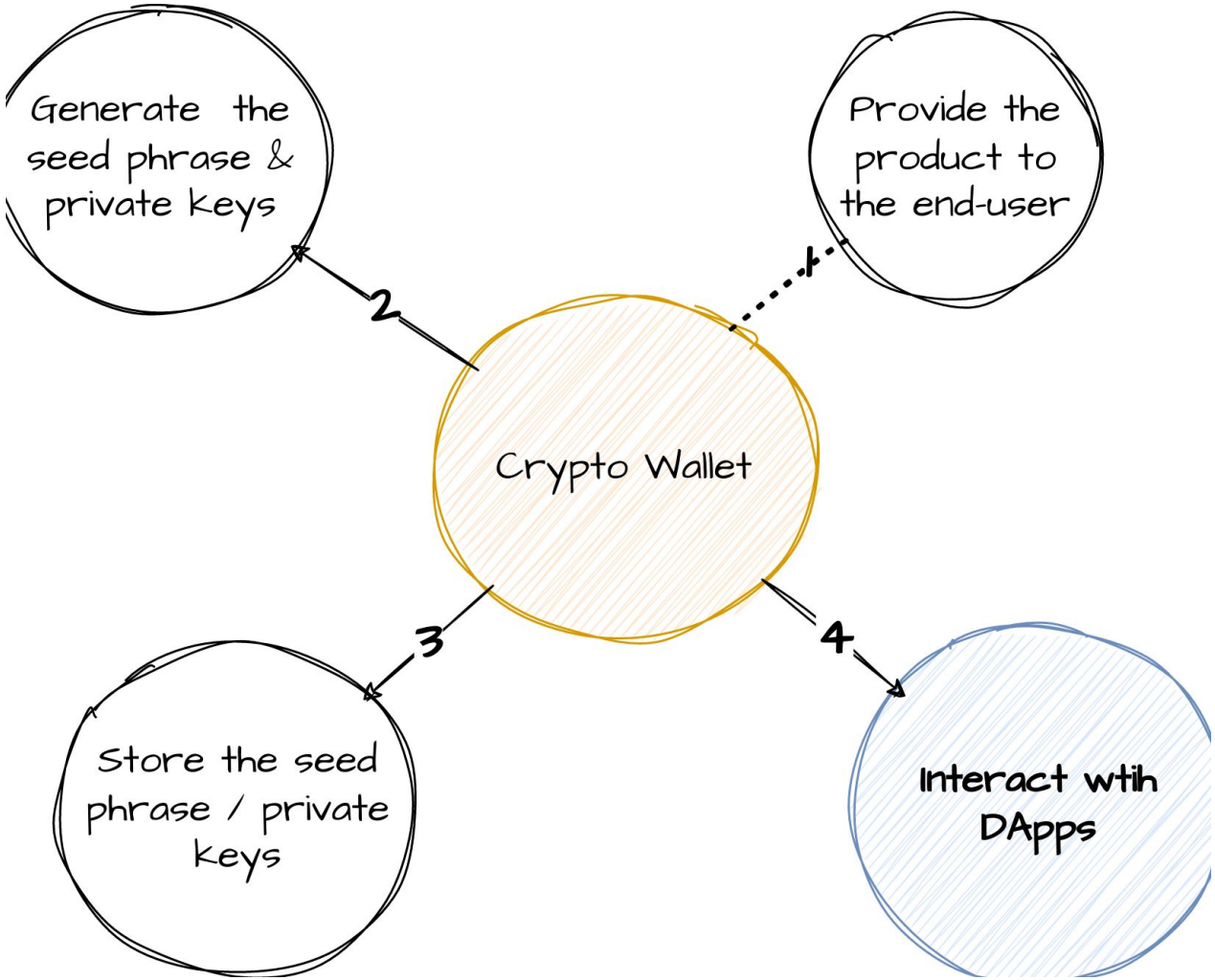

Random Salt
= Trezor Storage
+ Hardware Salt
KEK
Choose & Enter
PIN
PBKDF2
KEIV
User
Authenticated Encryption
with Associated Data
(AEAD)
ChaCha20Poly1305
Data Encryption Key
Encrypted Data Encryption
Key
Tag

Trezor - Kraken - 2020

- Fault-injection attack to extract the entire flash-contents
- Trezor firmware uses an encrypted storage
  - Developed a script to crack the PIN of the dumped device
  - The script was able to brute force any 4-digit pin in under 2 minutes.
- A PIN brute-force is not directly possible directly on the the device, because the storage is automatically wipes after 16 unsuccessful attempts.

[Kraken Identifies Critical Flaw in Trezor Hardware Wallets]

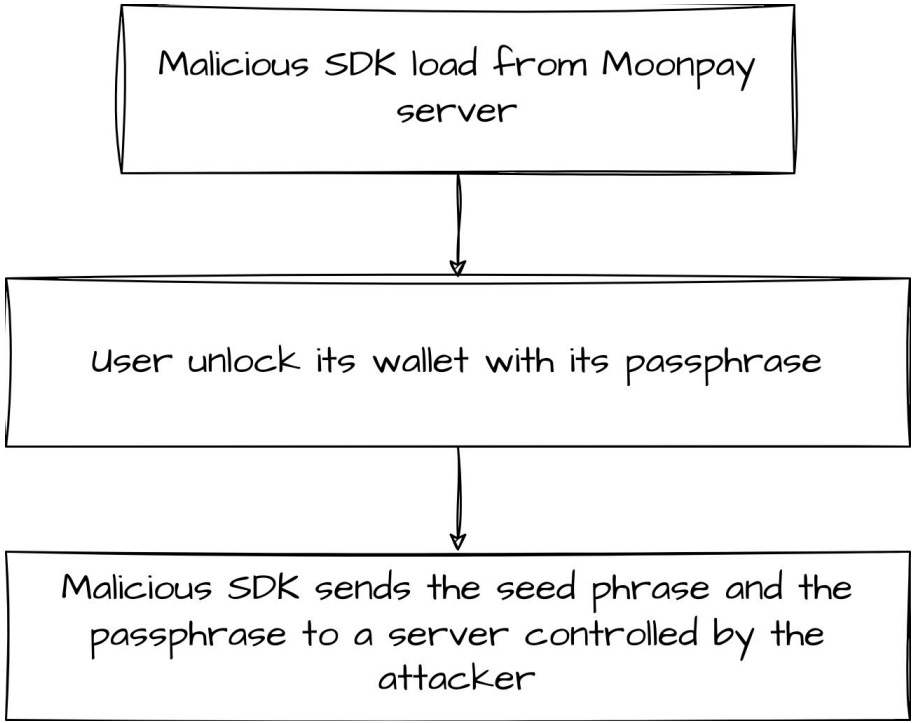



Generate the seed phrase & private keys
Provide the product to the end-user
1
2
Crypto Wallet
3
4
Store the seed phrase / private keys
Interact wtih DApps

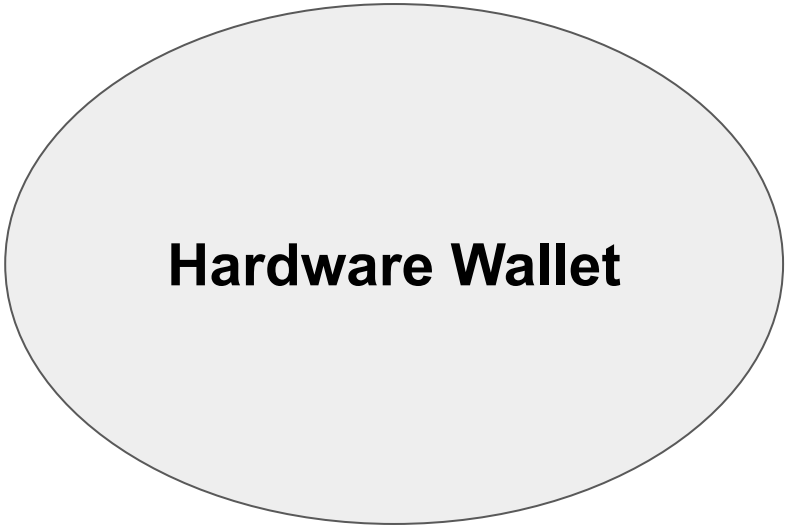

Software Wallet

IOTA Supply chain attack - 2020

- Trinity was a software wallet (Desktop and mobile phone) managed by the IOTA foundation
- Moonpay is a third-party service that allows users to purchase IOTA tokens from within the Trinity wallet.
- On 2020 the Trinity Wallet was attacked via a third-party dependency from Moonpay

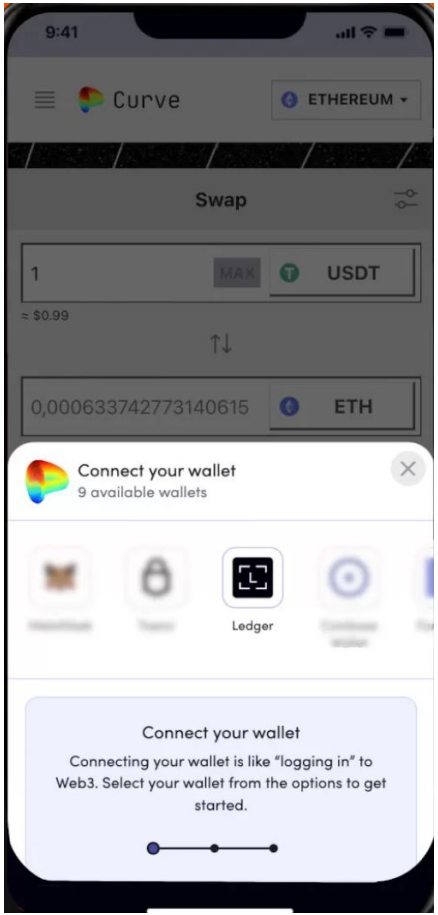


IOTA Foundation - Medium


Hardware Wallet

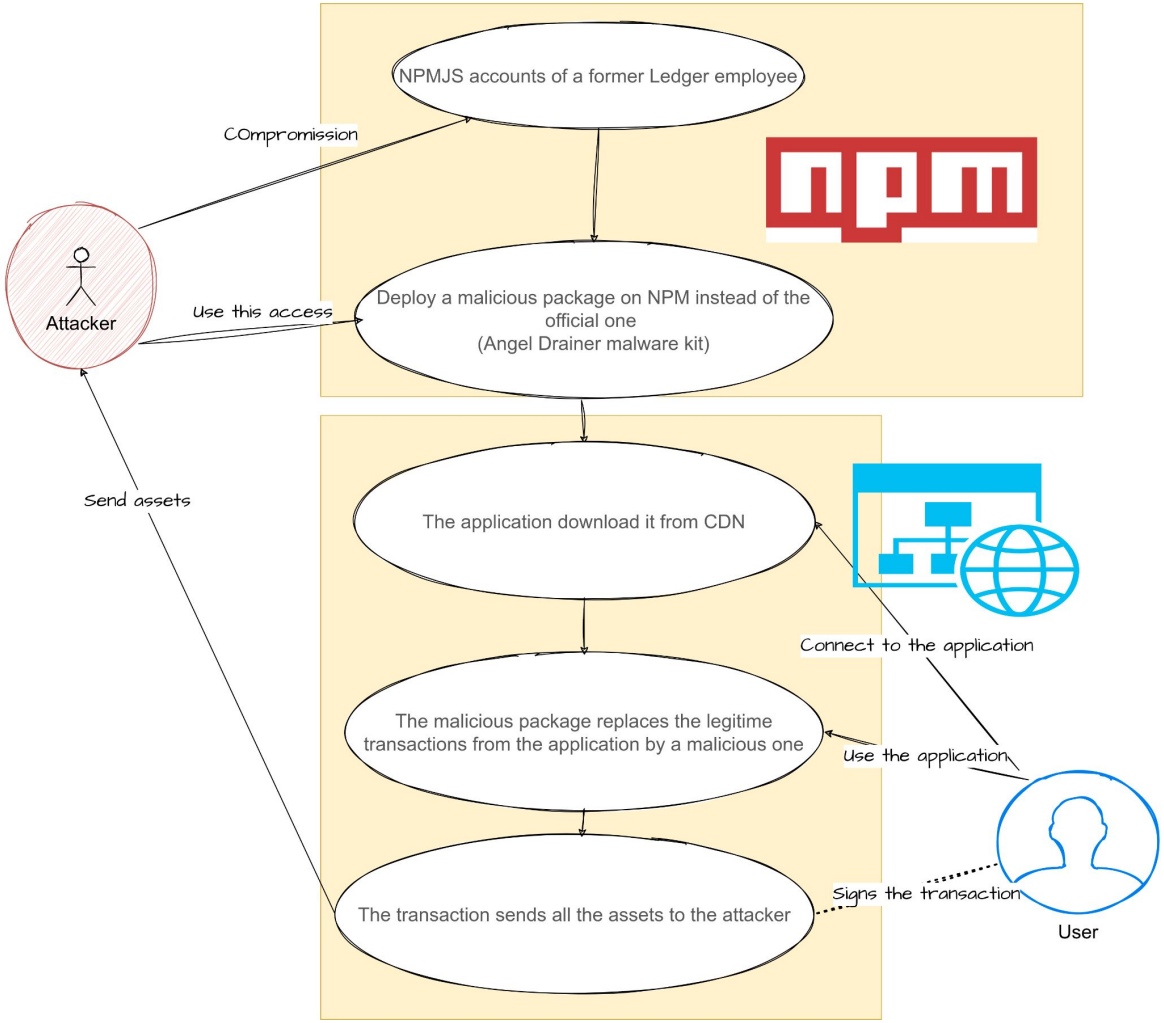
Ledger Connect Kit - 2023

- Ledger offers Connect Kit, a javascript library, to quickly install a Ledger button on a DApp
- The library is available through NPMJS
- Several DApp don’t store the package inside the front-end app but download the latest version available from a CDN.
- What happens?



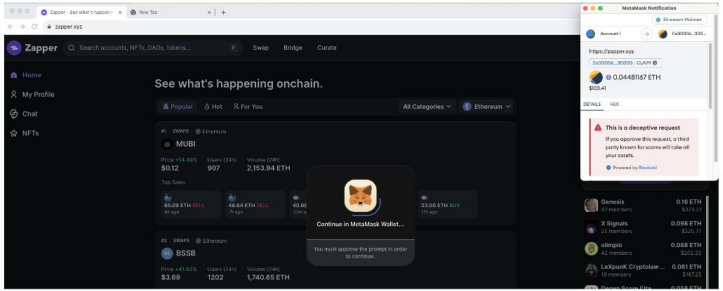
[Ledger - Security Incident Report]

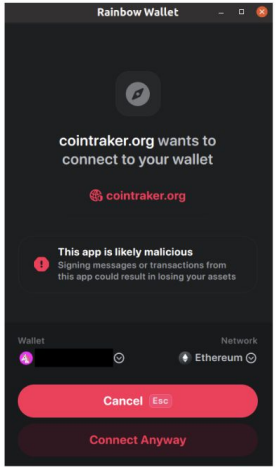
[Ledger - Connect Kit]

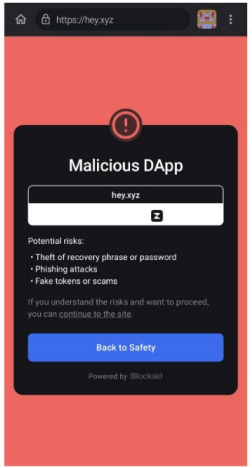

NPMJS accounts of a former Ledger employee
COmpromission
Attacker
Use this access
Deploy a malicious package on NPM instead of the official one
(Angel Drainer malware kit)
Send assets
The application download it from CDN
Connect to the application
The malicious package replaces the legitime transactions from the application by a malicious one
Use the application
The transaction sends all the assets to the attacker
Signs the transaction
User

Detection and accident response

- Quickly detected by different tools and users.
- Ledger quickly replaced the malicious version on NPM (40 minutes).
- With the caching mechanism in CDNs, the malicious package remained active longer.

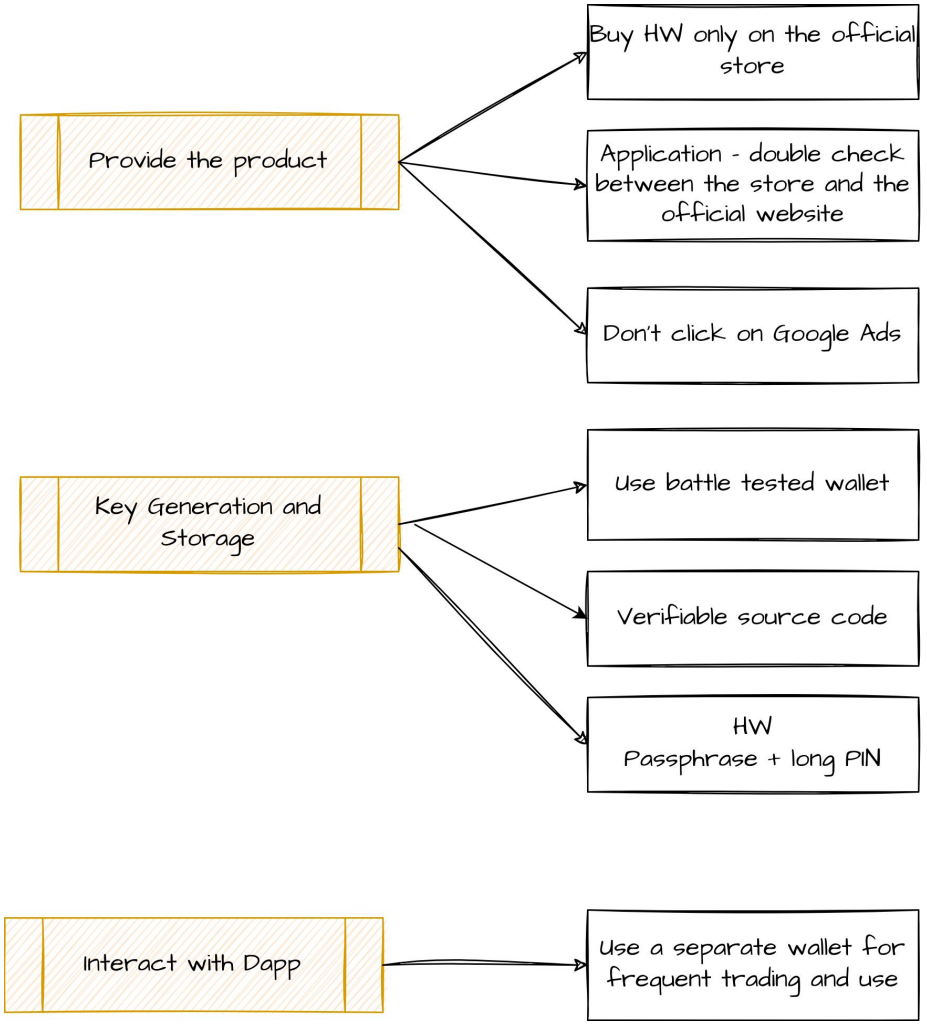






[blockaid - Attack Report: Ledger Connect Kit]

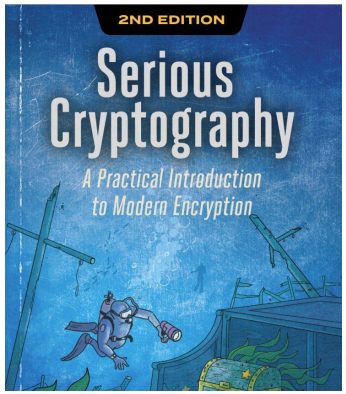
What can you do as a user ?



Thanks

Reviewers:

- Jean-Philippe Aumasson
- Taurus employees



Special thanks to all the organisers of Black Alps !