

Keybleed: Attacking The OneKey Mini

And other tidbits from the archives of Team Kairos

Eric Michaud/Tom Smith- Co-Founder/Partners



UNCIPHERED

CRYPTOCURRENCY RESCUE

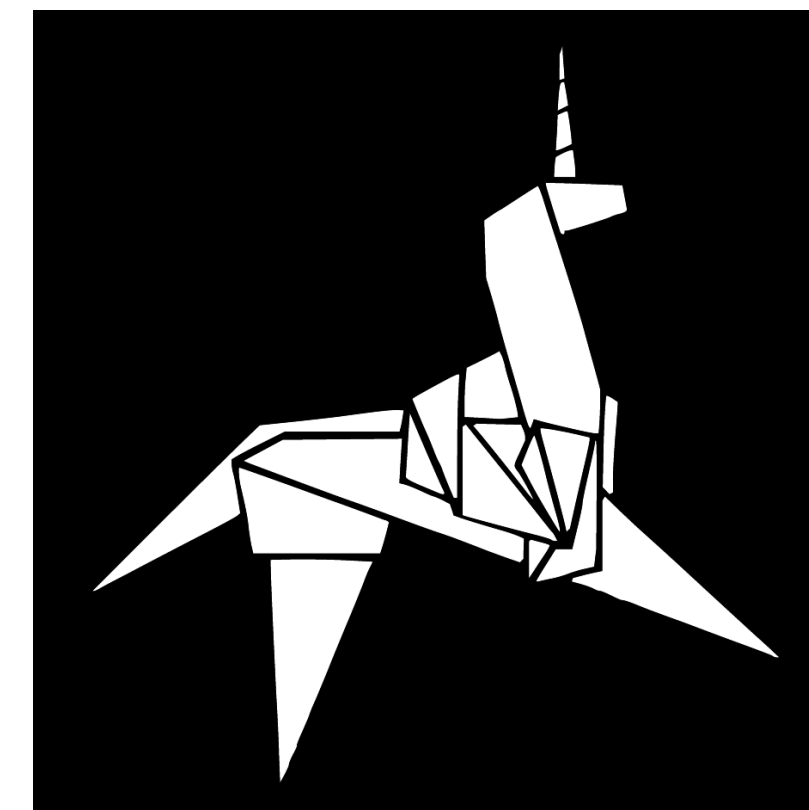


Who is Unciphered?

- Who is Eric?
 - Previously
 - TOOOL.us/ANL/PSOne/Rift Recon
 - Ultramarathoner
 - Macgyvers' Union member 554
- Who is Tom Smith
 - CTO
 - 20 yr HPC/FPGA/Glitching SME
- But....really... Who is Unciphered, Team Kairos, and Team Goolickers?
- The world's first and only Institutional Cryptocurrency Rescue & Recovery company.
- Started in February 2021 with a full and reliable exploit on the current Trezor One firmware and we didn't stop there!

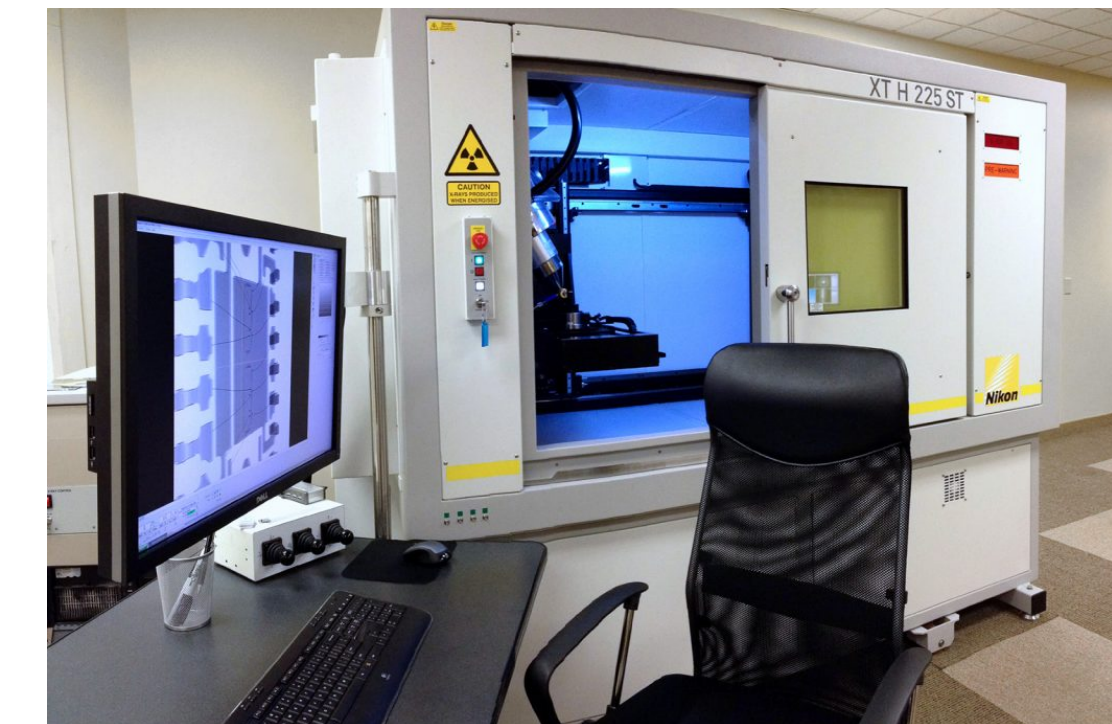
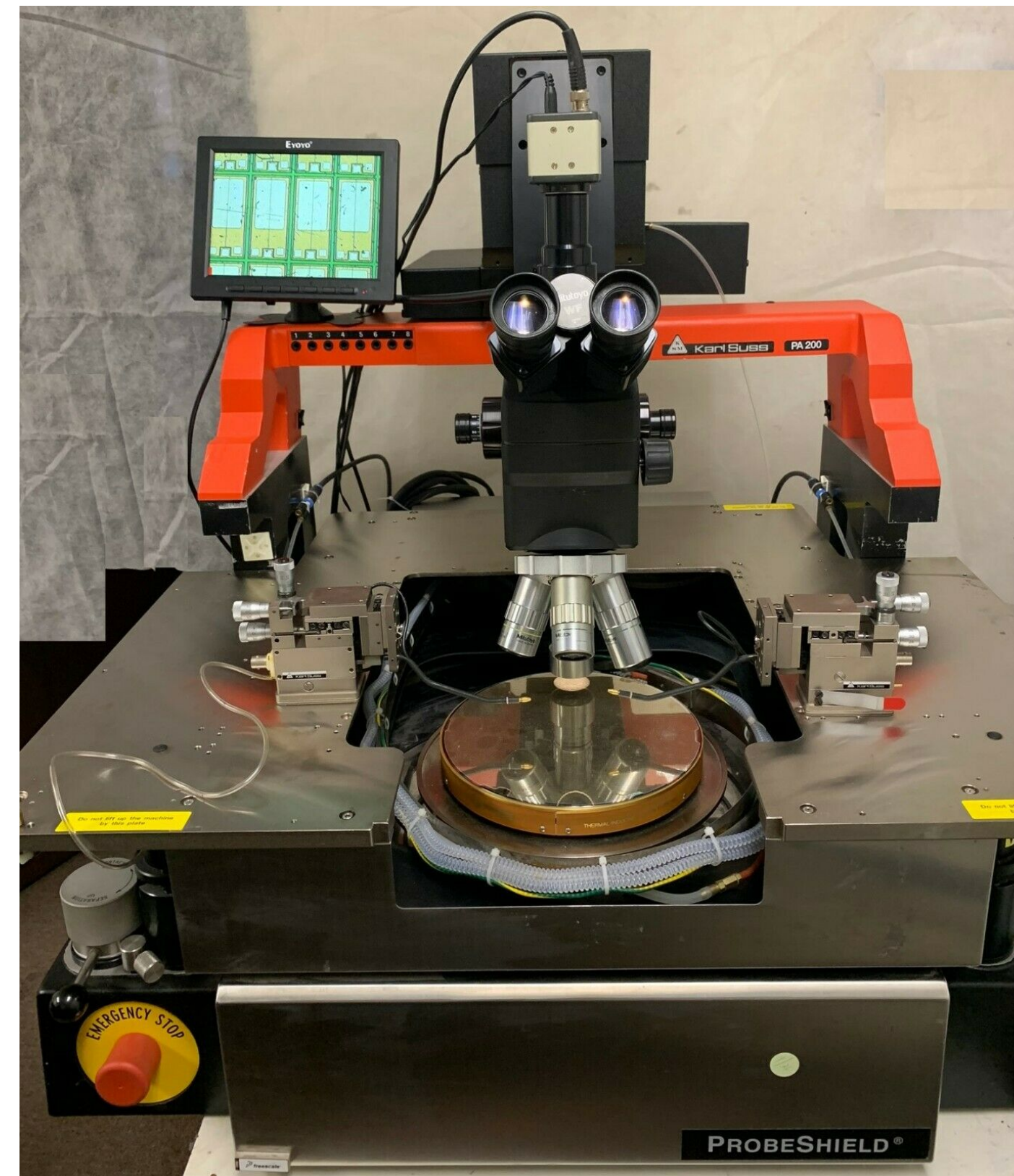
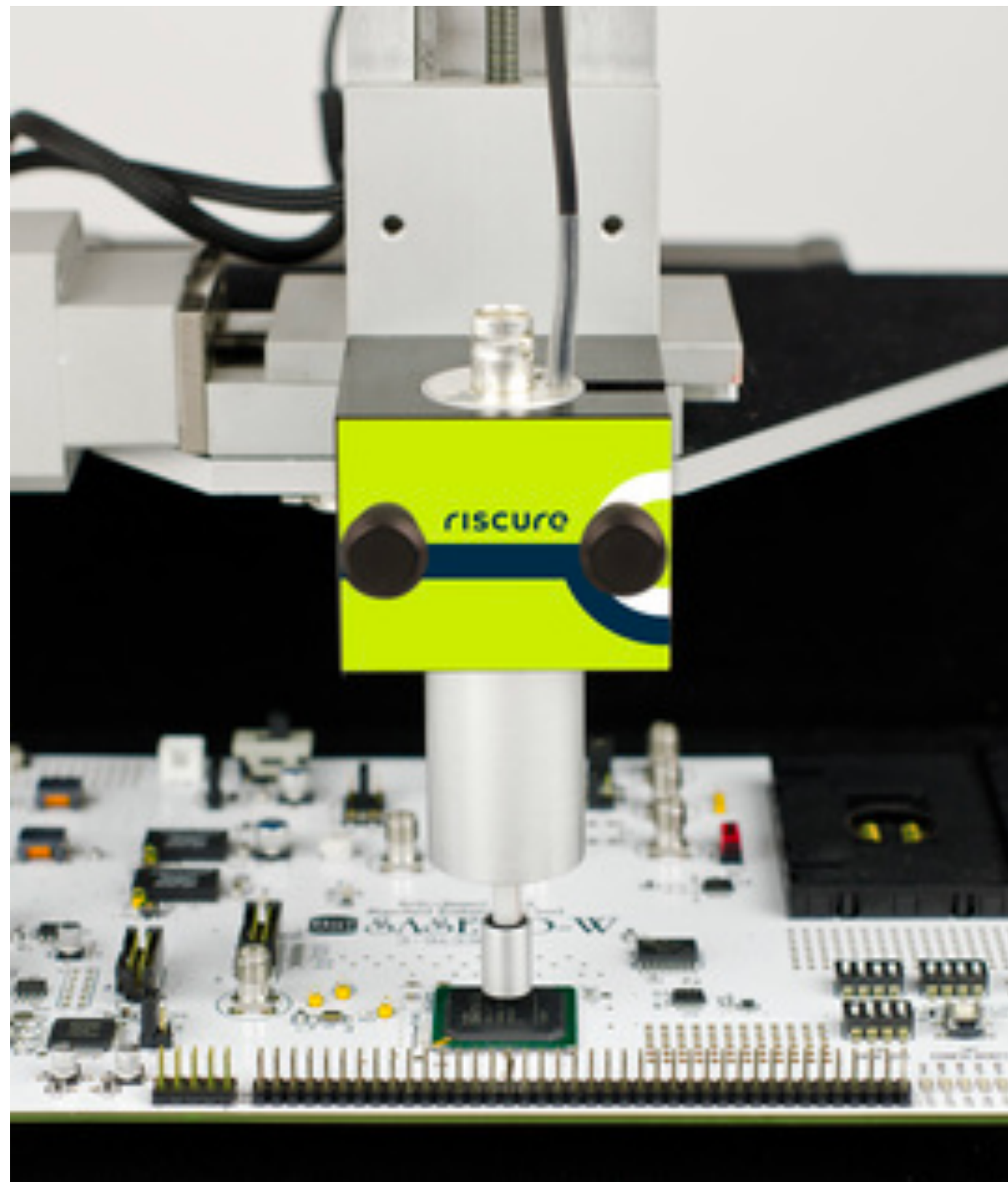


UNCIPHERED
CRYPTOCURRENCY RESCUE



Sample of Hardware Lab

No tours. Sorry!



Sample of Hardware Lab

BUUUUTT WAIT!!! WE JUST POSTED OUR FIRST EE LAB TOUR!



“Eventually on a long enough time horizon every crypto wallet becomes vulnerable. ~@WeldPond / L0pht”

Today we talk about miracles



UNCIPHERED
CRYPTOCURRENCY RESCUE



UNCIPHERED
CRYPTOCURRENCY RESCUE



UNCIPHERED
CRYPTOCURRENCY RESCUE



UNCIPHERED
CRYPTOCURRENCY RESCUE

Not this miracle

(But do look up B-Fields and Magnetic Flux Density if you were curious.)



UNCIPHERED
CRYPTOCURRENCY RESCUE



UNCIPHERED
CRYPTOCURRENCY RESCUE

Miracles = Exploits

Our customers think they are miracles



UNCIPHERED
CRYPTOCURRENCY RESCUE



UNCIPHERED
CRYPTOCURRENCY RESCUE

The Saint

- Val Kilmer played Simon Templar a reformed thief
- He states a line in the film that to become a Catholic Saint you must perform 3 miracles
- Today we're going to show 5 😅 miracles
 - Or exploits



Cold Storage Coin

- Remember Casascius Coins?
- Preloaded with Crypto
- Had to shut down because it was classified as a financial instrument
- Defeated with friends at DefCon
- https://en.bitcoin.it/wiki/Casascius_physical_bitcoins
- Now they come unloaded





UNCIPHERED
CRYPTOCURRENCY RESCUE



UNCIPHERED
CRYPTOCURRENCY RESCUE



OUR OFFICIAL RELEASE IS ON ITS WAY



A beta is available, but we caution users that bugs may still exist and formats may change.

We recommend small amounts only, and remind you that use of this software is at your own risk.

Launch Beta

Ethereumwallet.com

- A Non-Custodial wallet system written in Javascript, based on Kryptokit
 - Ran from 2016-2017 - Beta started in January 2016
- But we cracked a wallet from 2015 using the same techniques
 - May have been beta testing.
 - If anyone has more data on that please let us know.
 - First seen on wayback jan 10 2016.
- We have recovered **142*** mostly previously funded private keys on Legacy to date
- Legacy wallets were generated using a fast_sha256 package that expected input to be passed as a buffer but a string was used.
 - This had major impact on the expected keyspace of $\sim 2^{179}$. 98% of the keys now resided in a space no larger than 2^{33} .
- There are 4 types of Ethereumwallet.com wallets.

ethereumwallet.com

- Legacy ('c' or 'p' prefix for url frag) otherwise 'd' or 'q'
- Prefix and length indicate if a passphrase was used
- Improper use of Javascript buffers caused earlier versions of their code to force encode a script salt as UTF-8 causing bytes above 0x7f to the three byte replacement characters.
- This greatly reduced the entropy of the salt.
 - When an updated dependency fixed this issue, any previously created wallets could no longer be replicated via the site.

```
→ ethereumwallet ./ethereumwallet -frag qwfVrEv3uMUPIEHhcTU2FY4IGm79s3pc9f4abd63b00c54d -se < pswd.lst | jq
```

I



UNCIPHERED
CRYPTOCURRENCY RESCUE

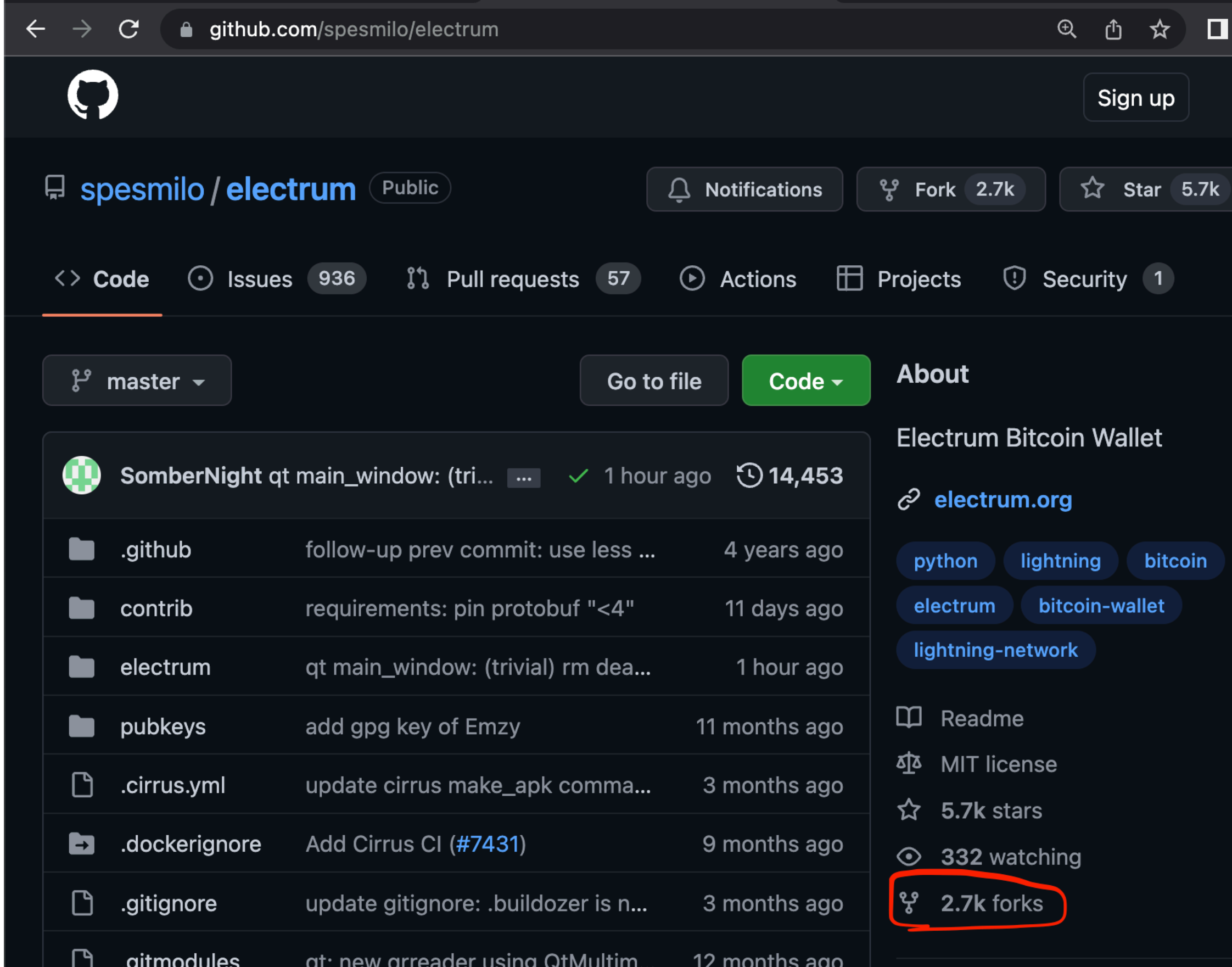


UNCIPHERED
CRYPTOCURRENCY RESCUE

Electrum Wallet



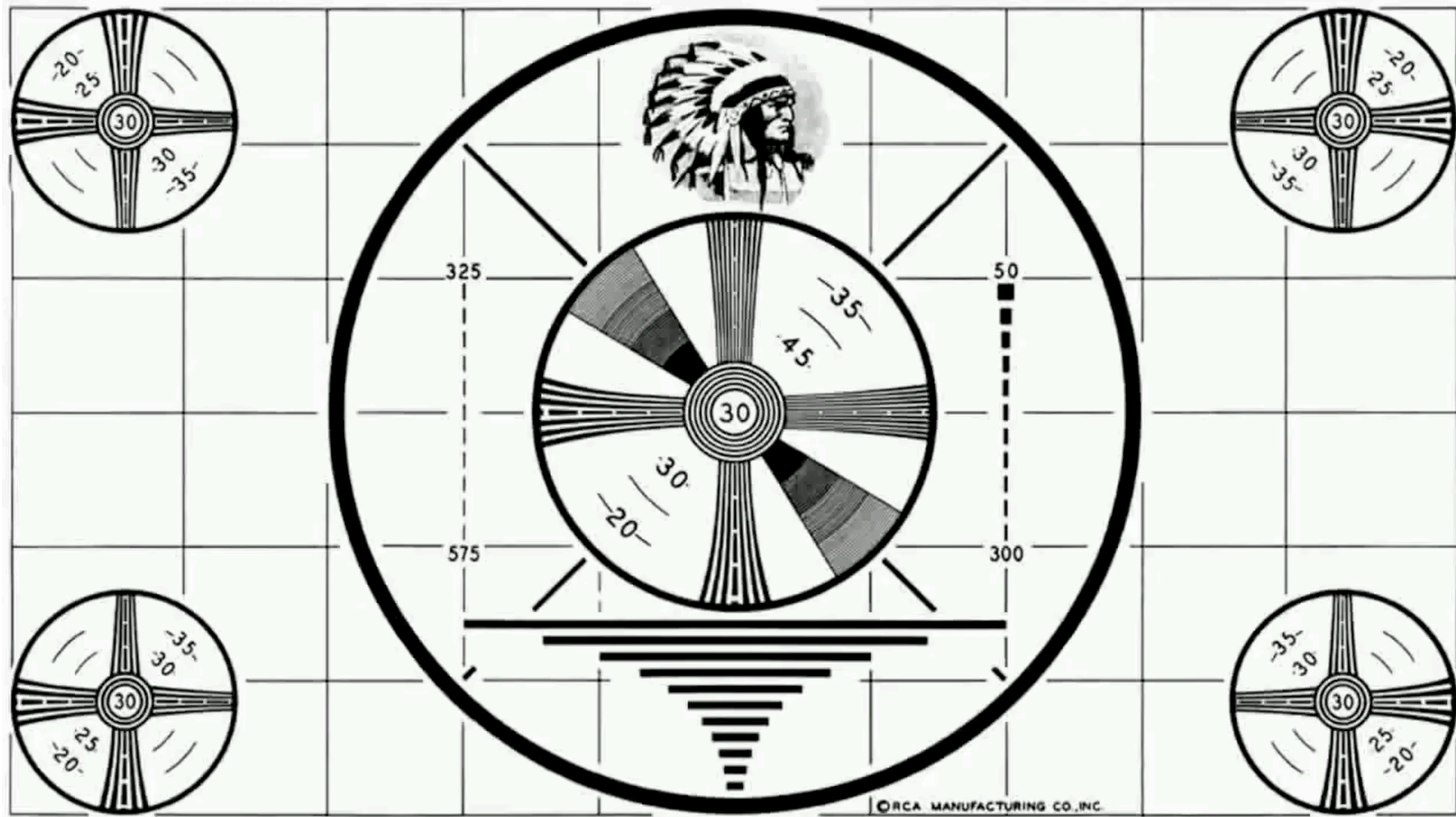
The screenshot shows the Electrum Bitcoin Wallet website. At the top, there is a navigation menu with links for Home, Download, Documentation, Community, and About. A warning message states: "Warning: Electrum versions older than 3.3.4 are susceptible to phishing. Do not download Electrum from another source than electrum.org, and learn to verify GPG signatures." Below this is a large image of ice cubes with a text box that reads: "Cold Storage: Sign transactions from a computer that is always offline. Broadcast them from a machine that does not have your keys." At the bottom, there are six feature icons: Safe (private keys encrypted), Forgiving (funds recoverable from a secret phrase), Instant On (fast due to servers), No Lock-In (private keys can be exported), No Downtimes (decentralized servers), and Proof Checking (transactions verified using SPV).



The screenshot shows the GitHub repository for spesmilo/electrum. The repository is public and has 5.7k stars and 2.7k forks. The repository contains several files and folders, including .github, contrib, electrum, pubkeys, .cirrus.yml, .dockerignore, .gitignore, and gitmodules. The 'About' section on the right lists the repository as 'Electrum Bitcoin Wallet' with a link to electrum.org and tags for python, lightning, bitcoin, electrum, bitcoin-wallet, and lightning-network. The '2.7k forks' statistic is highlighted with a red circle.

Electrum Wallet

- While we built a PoC demo on windows, Electrum Wallet is a cross-platform product.
- The failure to sanitize the QR code exists on all platforms.
- It is possible that this is exploitable on other platforms, however, we have not looked into it.
- It took 1 minute 35 seconds - Authentication token capture
 - There are various techniques for abusing windows authentication tokens.
 - Obviously, in 2022 with Net-NTLMv2 this isn't the same as LANMAN hash malfeasance.
 - However, you can still attack the token to get something you can abuse.



Electrum Wallet Win10 Wallet Steal PoC Rough Cut - CVE-2022-31246



UNCIPHERED
CRYPTOCURRENCY RESCUE



UNCIPHERED
CRYPTOCURRENCY RESCUE

All these projects were great to work with for the Responsible Disclosure

- Electrum Wallet - 2.1 - 4.5.1 - Windows/OSX/Linux/Android/Tails
- Electron Cash Wallet - 2.9.2 - 4.2.9 - Windows/OSX/Linux/Android/iOS
- Electrum-LTC - 2.1 - 4.5.1 - Windows/OSX/Linux/Android/iOS
- Electrum-Atom - 2.1 - 3.2.3 - Windows/Linux
- Electrum-SV - 1 - 1.3.13 - Windows/OSX
- Electrum-RVN - 2.1 - 4.5.1 - Windows/OSX
- Electrum-Zcash - 2.1 - 4.1.2.1 - Windows/Mac

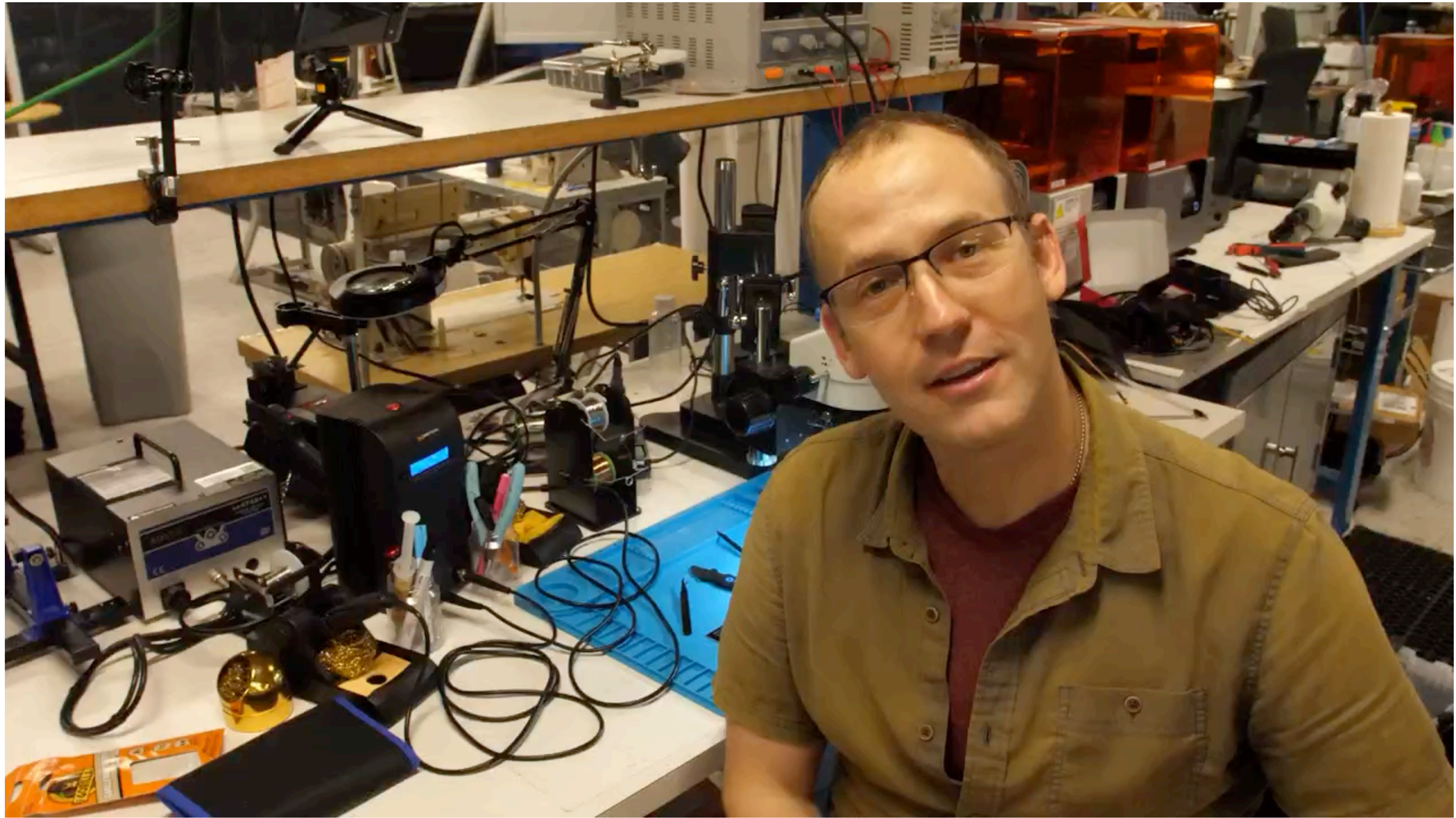
Big-Ups to ImmuneFi and Kraken Security Team

- Helped us get this issue fixed on the other 7 affected projects.
- As you guys are no doubt aware, vuln disclosure can be a painful process.
- So we're grateful for the assistance and coordination when dealing with this many projects.

Trezor Hardware Wallets

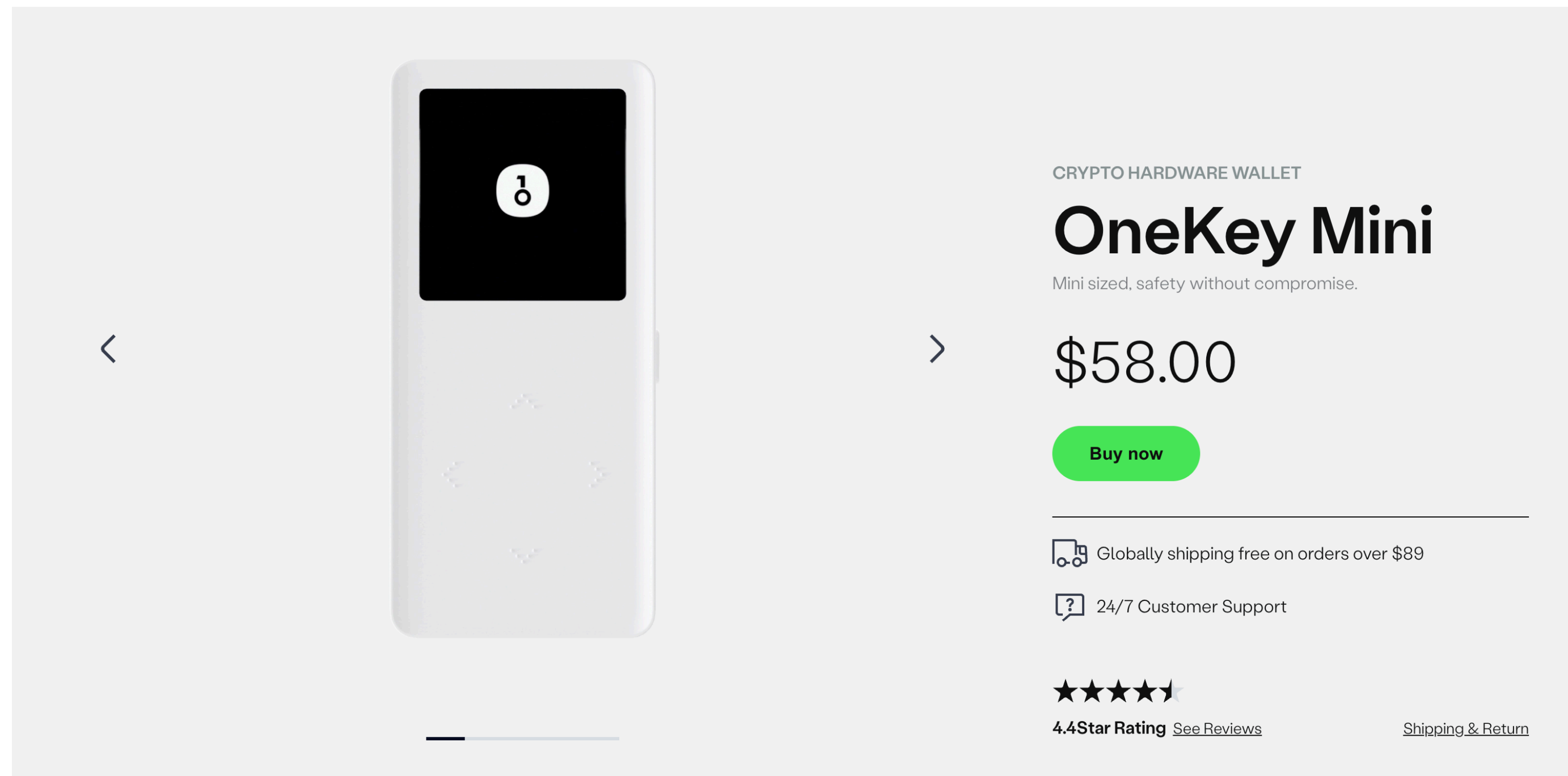
*“The Most Trusted Cold Storage for Bitcoin, Ethereum, ERC20 and Many More.”
- Trezor Amazon Copy*

We've all seen everyone use the Kraken or three year old 1day yeah?



OneKey - Hardware Wallets

"Open source crypto wallet trusted by millions."- OneKey Site Copy




CRYPTO HARDWARE WALLET


OneKey Mini

Mini sized, safety without compromise.

\$58.00

[Buy now](#)

 Globally shipping free on orders over \$89

 24/7 Customer Support

★★★★★
4.4Star Rating [See Reviews](#)

[Shipping & Return](#)

OneKey - Hardware Wallets

Expected Behavior

Physical access to the device should not lead to compromise of seeds.

Current Behavior

There are two behaviors that when combined give an attacker the mnemonics/seed. First, we extract the protect key, and second, we decrypt the seed using that key.

First, during bootup, the firmware checks whether the SE has been provisioned or not by reading the SE configuration.

```
// atca_api.c

void atca_config_init(void) {
...
    atca_assert(atca_read_config_zone((uint8_t *) &atca_configuration),
                "get config");
}
```

OneKey - Hardware Wallets

First it checks whether the lock is set:

```
if (atca_configuration.lock_value == ATCA_LOCKED) {  
    return;  
}
```

If it is not locked, it writes a “protect key” to the SE, without using any encryption:

```
for (int i = 0; i <= ATCA_KEY_ID_MAX; i++) {  
    if (!(atca_configuration.slot_locked & (1 << i))) {  
        continue;  
    }  
    ...  
    switch (i) {  
    ...  
    case SLOT_IO_PROTECT_KEY:  
        atca_assert(atca_write_zone(ATCA_ZONE_DATA, SLOT_IO_PROTECT_KEY, 0,  
0,  
                                pair_info->protect_key,  
ATCA_BLOCK_SIZE),  
                    "init IO key");  
        atca_assert(atca_lock_data_slot(SLOT_IO_PROTECT_KEY), "lock slot1  
");  
        break;  
    ...  
    }  
}
```

OneKey - Hardware Wallets

An attacker is able to make the firmware believe the SE has not been provisioned by doing a man-in-the-middle between SE and CPU.

By modifying the response from the SE, the CPU can be tricked into believing the `protect_key` has not yet been provisioned. Specifically, an attacker has to **change** `atca_configuration.lock_value` such that it does not equal `ATCA_LOCKED`, and **set the** `SLOT_IO_PROTECT_KEY` bit in `atca_configuration.slot_locked` to 0.

When those conditions are met, the CPU writes the `protect_key` to the SE.

OneKey - Hardware Wallets

In the second phase, we use the `protect_key` to get the seed. Normally, when accessing the seed, the CPU first asks for and verifies the user's PIN. When correct, it sends a command to the SE to read out the encrypted seed:

```
// se_atca.c

bool se_export_seed(uint8_t *seed) {
    uint8_t pin[32] = {0};
    pin_cacheGet(pin);

    atca_pair_unlock();
    if (ATCA_SUCCESS == atca_mac_slot(SLOT_USER_PIN, pin)) {
        if (ATCA_SUCCESS == atca_read_enc(SLOT_USER_SECRET, 0, seed,
                                         | pair_info->protect_key,
                                         SLOT_IO_PROTECT_KEY)) {

            return true;
        }
    }
    return false;
}
```

OneKey - Hardware Wallets

The slot is encrypted using a key stream generated by the following SHA-256 hash:

```
// atca_api.c
sha256_Update(&ctx, slot_key, 32);           // writekey
sha256_Update(&ctx, other_data, 4);         // other data
sha256_Update(&ctx, pair_info->serial + 8, 1); // serial number
sha256_Update(&ctx, pair_info->serial, 2);   // serial number
sha256_Update(&ctx, zeros, 25);             // zeros
sha256_Update(&ctx, temp_key, 32);          // tempkey
```

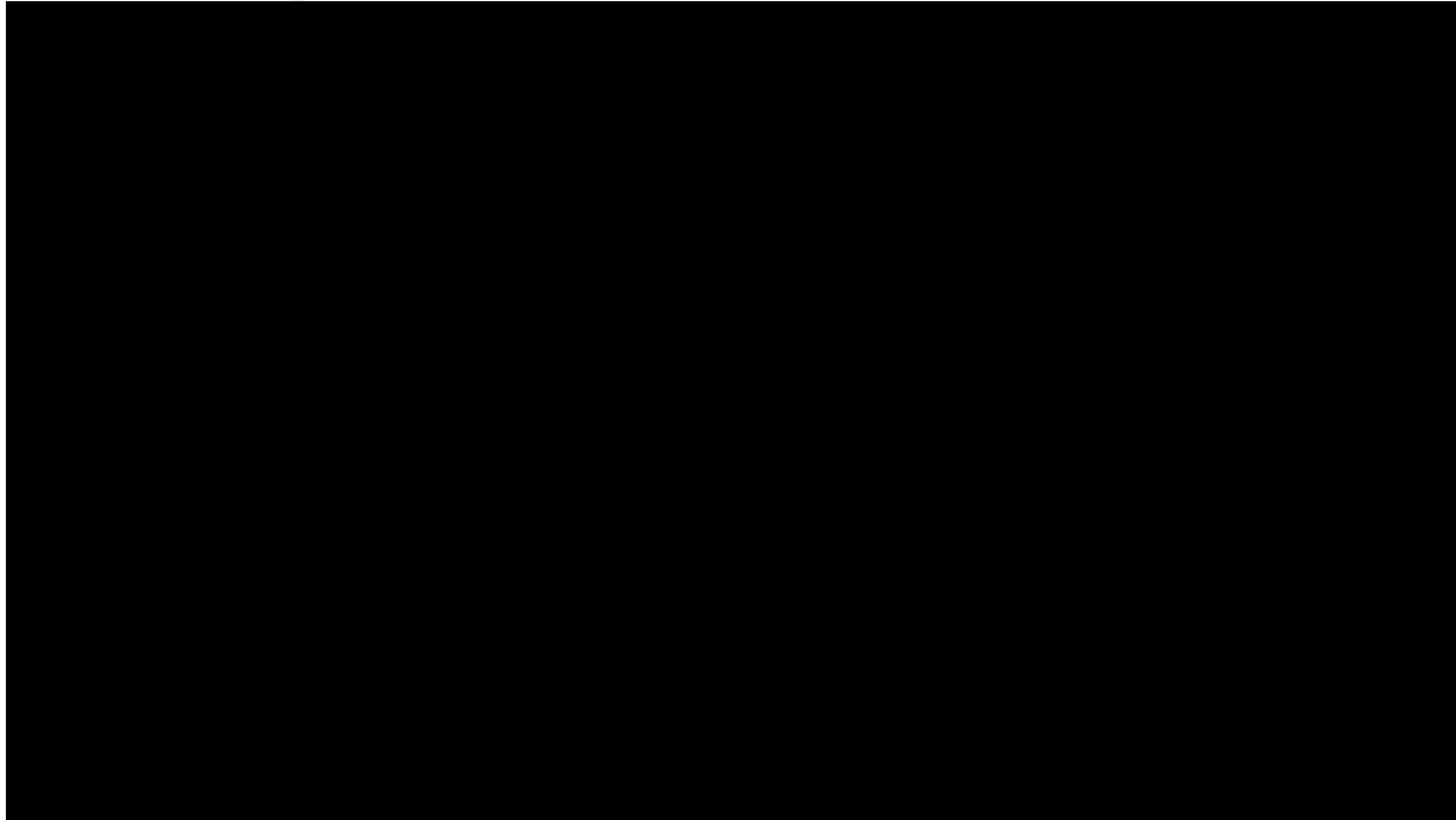
Since `slot_key` is in this case `protect_key`, and the serial number can also be obtained from `atca_configuration`, an attacker has everything needed to decrypt the slot containing the seed. No PIN is needed to send the `atca_read_enc` command, so after reading the response, decrypting the seed, it can be turned into the mnemonics of the wallet.

OneKey Hardware Wallets

Possible Solutions We Suggested

- Only allow provisioning once, e.g. by blowing a fuse in the CPU.
- Cryptographically tie the seed encryption to the PIN (does not protect against brute-force attacks)
- Use an SE that supports a cryptographic channel between CPU and SE, disallowing MITM
- Use SE features that require a specific password / PIN to unlock the zone where the seed is stored
- Seek independent certification services that cover hardware attacks such as MITM, but also perturbation attacks, side channel attacks, and invasive attacks.

OneKey 1 Second 1 Punch Attack



OneKey's Responsible Disclosure Program

Scope

This bug bounty program extends to all code within the [OneKey Github Repo](#).

Bounties for potential bugs include, but are not limited to:

- Private keys, storage, forensics
- Task, and CI/CD workflow vulnerabilities
- Domain hijacking, Secrets compromise
- Authorization and privilege issues

More generally, if it lives in the repository* and affects OneKey's security, it's fair game.

** There are some components of the OneKey repository that are not created by the OneKey team, but which still could be relevant to overall security. If a bug or exploit makes use of any external libraries or submodules, it will be considered on a case-by-case basis for eligibility.*

Rules

Submission Guidelines

All bugs reported must be done through the creation of an issue in the OneKey github repo, or *if the submitter wishes to disclose privately, or to remain anonymous* by an email sent to dev@onekey.so . Private submissions are still eligible for a bounty.

Unless there is a specific reason for a private disclosure, bugs should be submitted as issues on the OneKey GitHub repository, and tagged with the 'bug' label.

It is requested that all submissions follow the format defined in the [issue template](#) -- clarity of description and thoroughness of documentation will be a consideration for reward amount, in addition to impact and likelihood.

In the case of private bug disclosure, all relevant materials should be sent in email to `dev@onekey.so` -- and should follow the same template as a public issue.

Once submitted, the issue will be responded to, verified, accepted, and rewarded accordingly.

Submission Branches

Participants in the program are free to submit bugs on branches in the OneKey codebase:

- against the `master` branch which will be tagged as the mainnet release for deployment

Bug Severity and Bounties

In the same manner as the [Ethereum Bug Bounty Program](#), submissions will be evaluated by the OneKey team according to the [OWASP risk rating methodology](#), which grades based on both *Impact* and *Likelihood*.

It is at the *sole discretion of OneKey* to decide whether or not a bug report qualifies for a bounty, and to determine the severity of the issue

Severity levels:

- *Note*: Up to \$500 USD (min. \$100)
- *Low*: Up to \$1,000 USD (min. \$500)
- *Medium*: Up to \$2,000 USD (min. \$1,000)
- *High*: Up to \$5,000 USD (min. \$2,000)
- *Critical*: Up to \$10,000 USD (min. \$5,000)

Issues reported may or may not constitute a security risk for the OneKey contracts. A higher severity will be awarded to vulnerabilities submitted that could potentially result in either the loss of funds, or a situation in which the contracts arrive in an undesirable state that cannot be rectified through existing contract mechanisms, such as 'emergency mode' or through a network upgrade. However, all submitted bugs and vulnerabilities will be considered for prizes.

OneKey Hardware Wallets

Almost 7 days later.....

OneKey's Updated Responsible Disclosure Program

Submission Branches

Participants in the program are free to submit bugs on branches in the OneKey codebase:

- against the `master` branch which will be tagged as the mainnet release for deployment

Bug Severity and Bounties

In the same manner as the [Ethereum Bug Bounty Program](#), submissions will be evaluated by the OneKey team according to the [OWASP risk rating methodology](#), which grades based on both *Impact* and *Likelihood*.

It is at the *sole discretion of OneKey* to decide whether or not a bug report qualifies for a bounty, and to determine the severity of the issue

- P0: \$5,000 USD
- P1: \$2,500 USD
- P2: \$1,000 USD
- P3: \$500 USD
- P4: \$250 USD

Severity levels based on

- The severity of the bug.
- The likelihood that the bug will affect users.
- The responsibility of the researcher — did the researcher take destructive action or otherwise harm the functioning of our systems.
- The role of the researcher — was the researcher the first person to discover the bug, or is the bug based on some public information.
- How well the report was written and how easy it is to understand.

Issues reported may or may not constitute a security risk for the OneKey contracts. A higher severity will be awarded to vulnerabilities submitted that could potentially result in either the loss of funds, or a situation in which the contracts arrive in an undesirable state that cannot be rectified through existing contract mechanisms, such as 'emergency mode' or through a network upgrade. However, all submitted bugs and vulnerabilities will be considered for prizes.

OneKey's Updated Responsible Disclosure Program

Ineligible Bugs

Any vulnerabilities or flaws in other software tools created by OneKey (e.g. OneKeyJS, purser, tailor, etc.) are not eligible. Flaws in these software tools are welcome disclosures, but will not be awarded bounties for this bug bounty program.

When reporting vulnerabilities, please consider (1) attack scenario / exploitability, and (2) security impact of the bug. The following issues are considered out of scope:

- Attacks and vulnerabilities that depend on compromised keys or other security flaws outside the OneKey codebase (keyloggers, intercepted communications, social engineering exploits, etc.).
- Attacks that are accounted for in the system design, i.e. Ethereum network spamming, malicious reputation mining, malfeasance in OneKey administration.
- Critiques of the OneKey and overall mechanism design. We welcome suggestions and constructive criticism, and ask that it be directed to dev@OneKey.so.
- Clickjacking on pages with no sensitive actions
- Cross-Site Request Forgery (CSRF) on unauthenticated forms or forms with no sensitive actions
- Attacks requiring MITM or physical access to a user's device.
- Attacks requiring a compromised victim device.
- Previously known vulnerable libraries without a working Proof of Concept.
- Comma Separated Values (CSV) injection without demonstrating a vulnerability.
- Any activity that could lead to the disruption of our service (DoS).
- Content spoofing and text injection issues without showing an attack vector/without being able to modify HTML/CSS
- Rate limiting or bruteforce issues on non-authentication endpoints
- Vulnerabilities only affecting users of outdated or unpatched browsers [Less than 2 stable versions behind the latest released stable version]
- Software version disclosure / Banner identification issues / Descriptive error messages or headers (e.g. stack traces, application or server errors).
- Public Zero-day vulnerabilities that have had an official patch for less than 1 month will be awarded on a case by case basis.
- Tabnabbing
- Open redirect - unless an additional security impact can be demonstrated
- Issues that require unlikely user interaction
- Perceived security weaknesses without evidence of the ability to demonstrate impact (e.g. Missing best practices, functional bugs without security implications, etc.)

OneKey's Updated Responsible Disclosure Program

- Cross-Site Request Forgery (CSRF) on unauthenticated forms or forms with no sensitive actions
- Attacks requiring MITM or physical access to a user's device.
- Attacks requiring a compromised victim device.

**"THAT IS OUT
OF SCOPE"**
-said no attacker ever-



Ledger DonJon(Dungeon) On OneKey

OneKey Permadeath on the ATECC608 & STM32F405 Chipset?

*Talk: OneKey is all it takes: The Misuse of Secure Components in Hardware Wallets
@ Hardware.io*

Michael Mouchous & Karim Abedellatif

I asked for CVE's but they didn't file them yet.

CVE-2023-????? - Homemade Electromagnetic Fault Injection

CVE-2023-????? - Misconfiguration of PIN Auth Process

CVE-2023-????? - Laser Fault Injection Attack

<https://hardwear.io/usa-2023/speakers/michael-and-karim.php>

Ledger DonJon(Dungeon) On OneKey



NOTES ON DISCLOSURE

OneKey team has been contacted several times during second semester 2022

- Description of findings
- Recommendations

No proactive answer from them after 6 months

Design not modified since

Other products with same architecture


CONCLUSION

40-40

hardware.io
Hardware Security Conference and Training

keybleed.com

The Havebeenpwned for crypto wallets


KEYBLEED API SERVICES ABOUT  [Powered By Unciphered](#)

"Keybleed"

~A vulnerability or exploit that leads to key extraction from a software or hardware crypto wallet.

Check if your Cryptocurrency Wallet is Vulnerable to Known Exploits.

Powered by [Unciphered](#).



KEYBLEED Wallet Checker.

Submit your **Public Key** and our automated wallet checker will let you know if your wallet is vulnerable or becomes vulnerable in the future.

Name

Email

Email Confirm

Public Key

Unciphered Values

- **We're here to help**
- **Get back people what is lost - *Maximalists hate us***
- **Provide a standard on how Responsible Disclosure can be**
- **“Responsible Rescue”**

Responsible Rescue

- One more thing!
- <https://www.unciphered.com/responsible-rescue>
- Hypothetically, what happens when you can't patch ~1 Million Private Keys?
 - What happens when coins are from citizens in another country?
 - What happens when you have Nomad/Solana scale \$, \$, ETH, € at risk?
- Patch Tuesday doesn't work,
- CFAA - Computer Fraud and Abuse Act → Prison Time

Responsible Rescue

- **These are the questions are are asking right now.**
- **Coordinated doesn't disclosure doesn't fall into this.**
- **We don't have the answer yet.**
- **Contact me eric@unciphered.com**

Big Thanks To Team Kairos For the h4x you saw

- **Tom Smith**
- **Jerry**
- **Zonk**
- **Frank**
- **Max Cohen**
- **Tony Beretta**
- **Vyrus**
- **And a cast of many more**

Links

- Site: unciphered.com/
- Blog: unciphered.com/blog
- Engineering: unciphered.com/kairos
- Keybleed.com: <https://www.keybleed.com>



UNCIPHERED

CRYPTOCURRENCY RESCUE

<https://www.unciphered.com/letsgo>