Memory FORESHADOW: Memory FOREnSics of HArDware cryptOcurrency Wallets – A Tool and Visualization Framework

By
Tyler Thomas (University of New Haven (UNHcFREG)), Mathew Piscitelli (UNHcFREG), Ilya Shavrov (UNHcFREG), and Ibrahim Baggili (UNHcFREG)

From the proceedings of

The Digital Forensic Research Conference

DFRWS 2020 USA

Virtual -- July 20-24

Memory FORESHADOW: Memory Forensics of Hardware Cryptocurrency Wallets

University of New Haven's Cyber Forensics Research and Education Group

Samuel S. Bergami Jr. Cybersecurity Center

Tyler Thomas, Mathew Piscitelli, Ilya Shavrov, and Ibrahim Baggili

University *of* New Haven

UNHcFREG

This material is based upon work supported by the National Science Foundation under Grant No. 1921813. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation

University *of* New Haven

Who Are We?

- First semester graduate students at the University of New Haven
- Researchers at UNHcFREG
- Active members of the university's hacking team

University of New Haven

Agenda

- Research Problem & Motivation
- Contributions
- Literature Review
- Methodology
- Results
- Future Work
- Conclusion

Research Problem

- Given a memory image of a computer that was recently running a cryptocurrency hardware wallet client:

1. Can we detect the use of the wallet?
2. Can we extract forensically relevant data related to the use of the wallet (e.g. transaction history)?
3. If so, how long does the data persist in memory?

UNHcFREG

Contributions

- Primary account for memory analysis of cryptocurrency hardware wallet clients
- We publicly share our findings with the Artifact Genome Project (AGP)
- An open source tool for forensic analysis of cryptocurrency hardware wallet client memory
- Novel memory visualization framework for exploring memory persistence and integrity of artifacts

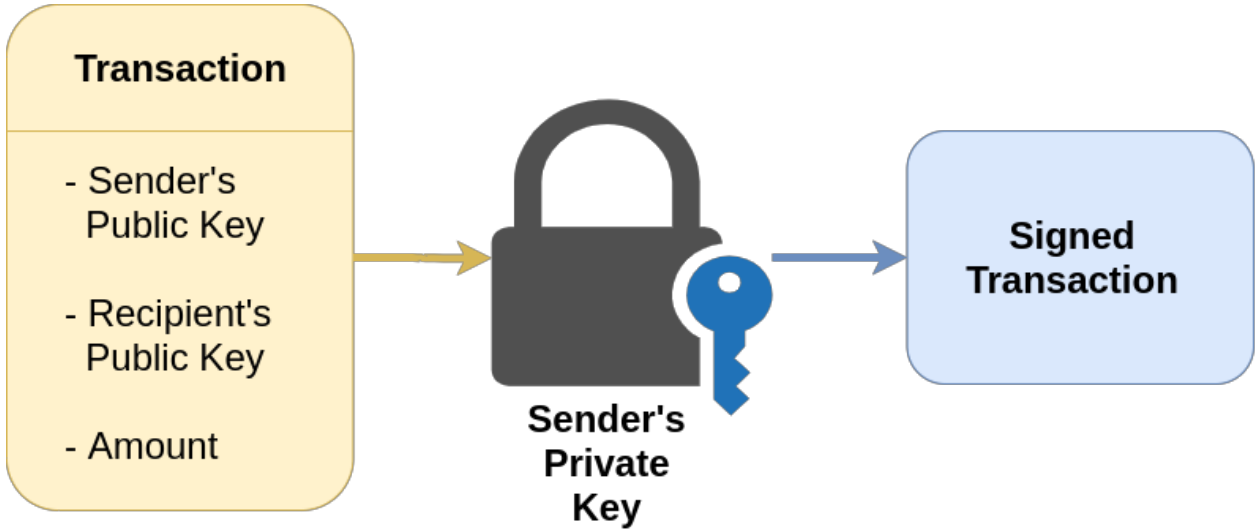
Literature Review

- Memory forensics has advanced rapidly since its inception in 2004
  - System level memory forensics (Volatility) [3]
  - Application specific memory forensics [4,5]
  - Differential analysis [6]

- Limited work has been done on cryptocurrency wallet memory forensics
  - Regex for public keys [7]
  - File system triage [9,10]
  - Network protocol forensics [1]
  - No focus on hardware wallet clients

Background: Cryptocurrency Review

- Decentralized currency
- Designed to preserve anonymity
- Public ledger of transactions
- Security lies in Public Key Infrastructure (PKI)



Background: Crypto Crime

UNHcFREG

TECHNOLOGY NEWS JANUARY 29, 2019 / 9:35 AM / A YEAR AGO

Cryptocurrency thefts, scams hit $1.7 billion in 2018: report

Gertrude Chavez-Dreyfuss 3 MIN READ

NEW YORK (Reuters) - Cryptocurrencies stolen from exchanges and scammed from investors surged more than 400 percent in 2018 to around $1.7 billion, according to a report from U.S.-based cyber security firm CipherTrace released on Tuesday.

How Cryptocurrencies Are Fueling Ransomware Attacks And Other Cybercrimes

Forbes Technology Council COUNCIL POST | Paid Program
Innovation

CNBC

Twitter hackers who targeted Elon Musk and others received $121,000 in bitcoin, analysis shows

On Wednesday, the Twitter accounts of some of the most famous people in the country were compromised as part of an apparent bitcoin scam.

3 days ago

Coindesk

Twitter Hack Used Bitcoin to Cash In: Here's Why - CoinDesk

Someone hacked Twitter Wednesday – and used bitcoin to capitalize on it. Why? Bitcoin is an alternative money system based on the value of ...

3 days ago

Decrypt

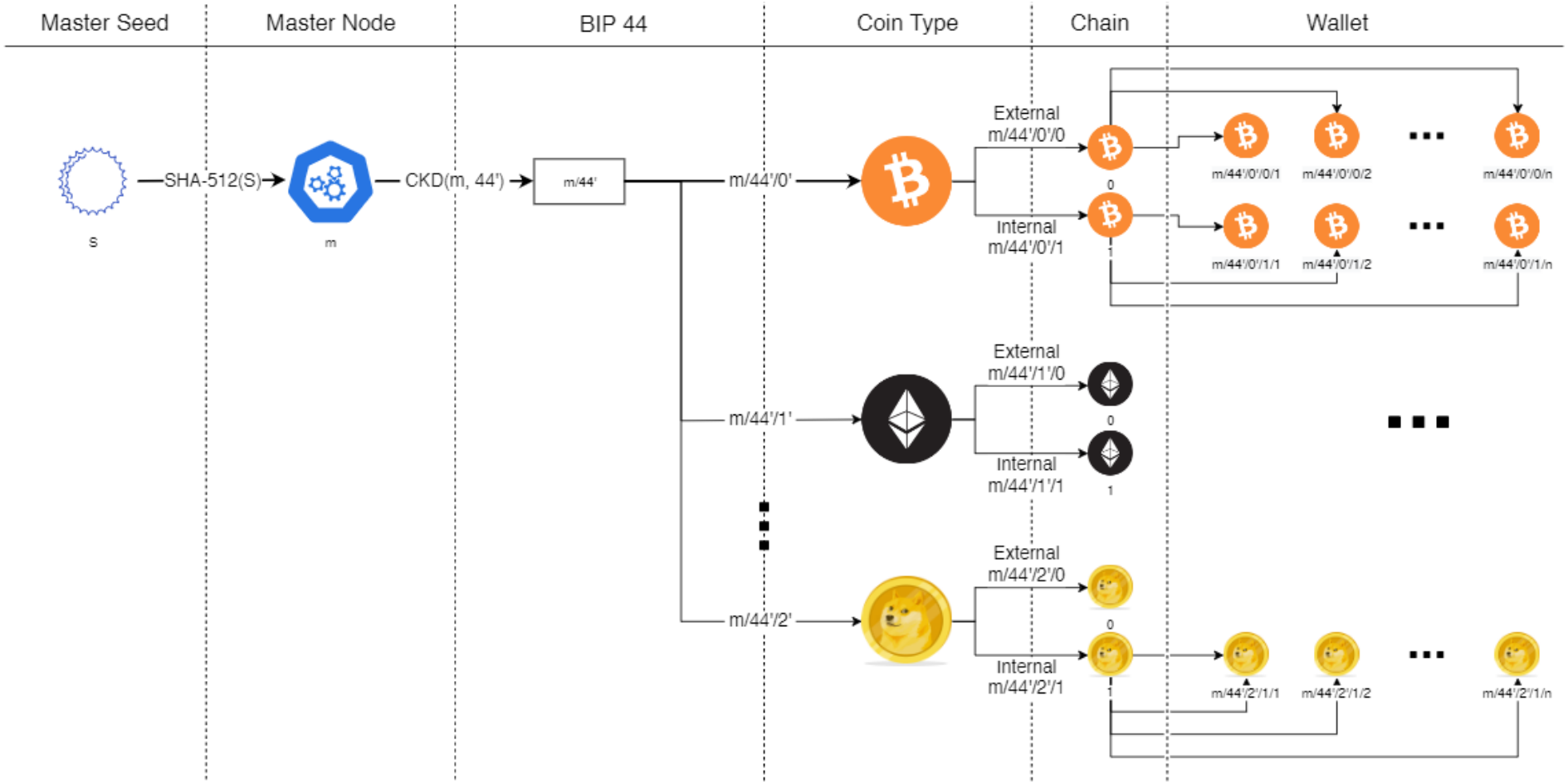
Stolen Bitcoin from Twitter hack is already being laundered: report

Blockchain analytics firm Elliptic reports that roughly 22% of the $120,000 worth of Bitcoin stolen in Wednesday's Twitter hack has been sent to ...

1 day ago

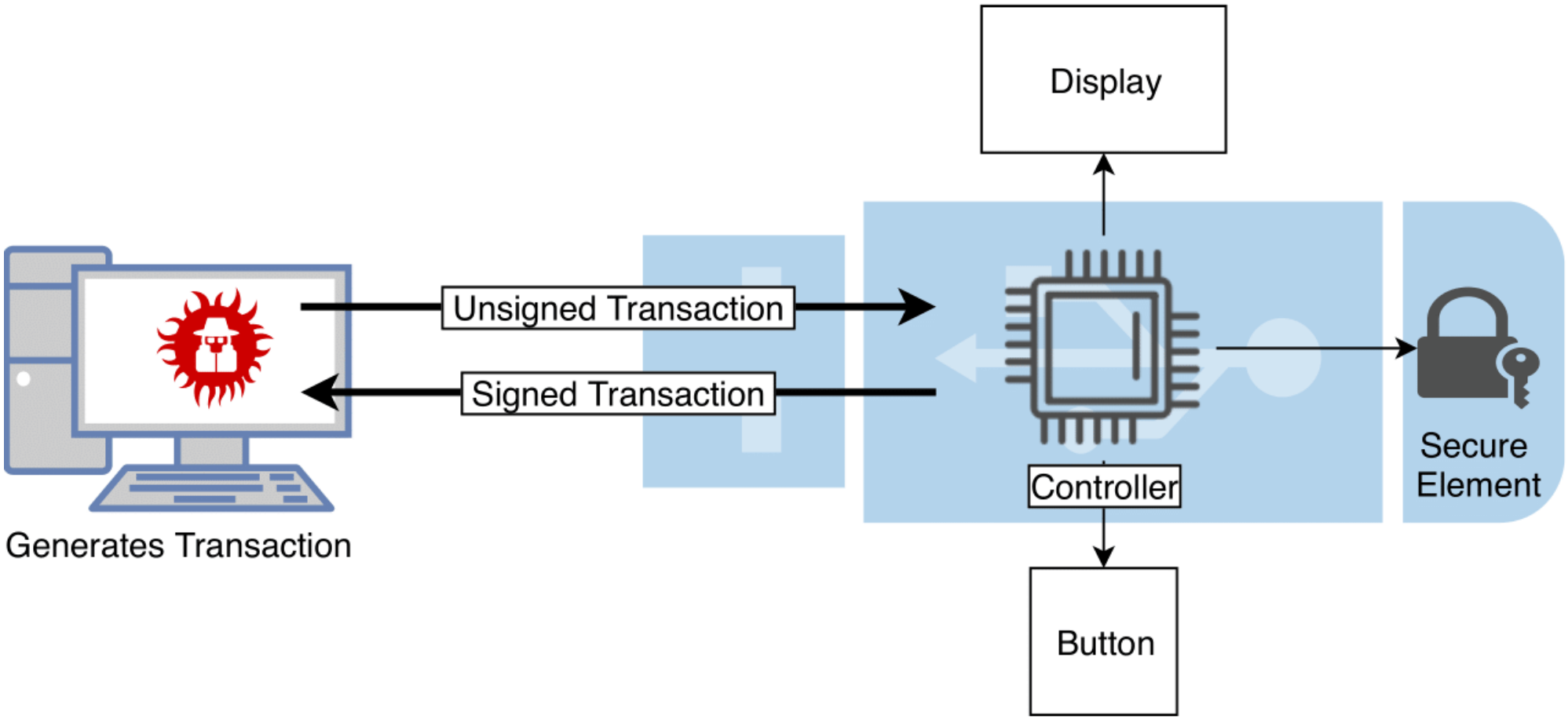
University of New Haven

Background: Hierarchical Deterministic Wallets



University of New Haven

Background: Why Hardware Wallets?



UNHcFREG

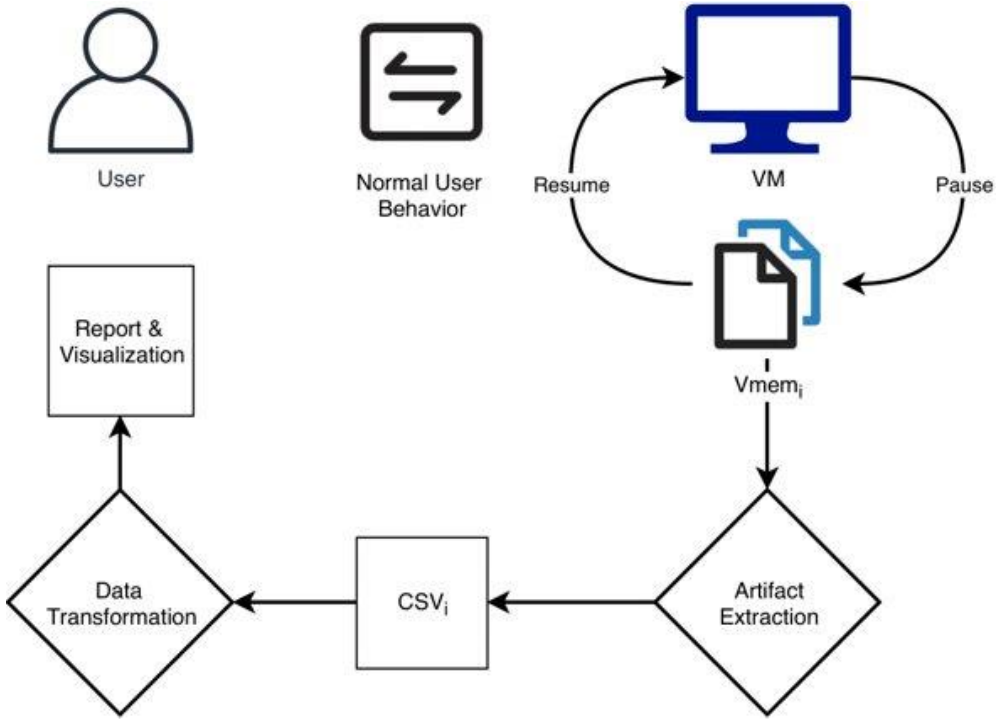
Software Clients Investigated

- Ledger Live (Ledger Nano X)
  - Desktop application built on Electron
  - JavaScript – Chromium V8 runtime
  - 1.3 million units sold[8]

- Trezor Wallet (Trezor Model One)
  - JavaScript browser application
  - Trezor Bridge daemon handles communication between USB device and browser
  - 800,000 units sold[2]

Methodology

1. Analyzed memory at runtime
   - Cheat engine
2. Automated the collection of memory dumps and the extraction of forensic artifacts
   - Volatility plugin
3. Created a framework to analyze memory and visualize findings



Apparatus

System Details		Application Details	
Device	**Details**	**Application Name**	**Version**
Processor	Intel Core i7-8750H	Cheat Engine	7.0
Operating System	Windows 7 Professional SP1 7601	VMWare Workstation Pro	15.5.1
System Type	64-bit OS, x64 processor	Mozilla Firefox	71.0
Virtual Memory (VRAM)	2.00 GB	Google Chrome	79.0.3945.88
Ledger Nano X	Firwmare 1.2.4-1	Ledger Live	v1.18.2
Trezor One	Firmware 1.8.3	Trezor Wallet	1.8.3
		Trezor Bridge	2.0.27
		Volatility	2.6

Artifact Extraction – Data Structures

- Primarily JSON based
- Electron IPC messages (Ledger Live)
- HTTP API requests

Data	Contents
LedgerLive	
— command-event	Transaction history and public keys
— JSON command	Transaction history and public keys
— ipc-message	Public keys
— JSON context	Device metadata
— API Request	Wallet addresses
— xpub	Public keys
Trezor Wallet (Firefox)	
— JSON Result	Transaction history
Trezor Wallet (Chrome)	
— API Request	Wallet addresses
— Passphrase	Extractable password
Trezor Wallet (Both)	
— JSON id	Device metadata and public keys
— xpub	Public keys

Artifact Extraction – Algorithm

Purpose:

- Identify structures in memory using YARA
- Serialize forensically relevant data structures from memory dump
- Repair structures that may have been damaged by memory smearing or deallocation

Structures

```
y ← yara.compile(rules)
s ← newlist                                   ▷ Serialized structures
for task ∈ processList do
    if task = LedgerLive.exe
            or task = Chrome.exe
            or task = Firefox.exe then
        for vad ∈ getProcessVad do
            m ← y.match(vad)                  ▷ Address of match
            if m = JSON_match then
                e ← end.match(vad) + offset
                data ← json.serialize(vad[m : e])
                s.insert(repair(data))
            end if
            if m = IPC_match then
                data ← vad[m : m + read_size]
                data ← data.asciiOnly.split()
                s.insert(data)
            end if
            if m = URL_match then
                data ← vad[m :].readUrlParams()
                s.insert(data)
            end if
            if m = XPUB_match then
                s.insert(vad[m : xpubSize])
            end if
        end for
    end if
end for
```

University *of* New Haven

Findings

Ledger Live

- Public Keys
- Transaction history
- Past and future addresses
- Device metadata

Trezor Wallet

- Public Keys
- Transaction history
- Past and future addresses
- Device metadata
- Unique device identifier
- Application password

Artifacts – Example Output (Ledger)

UNHcFREG

```
**********************************************
Found 3 public keys
.....Public key:
        libcore:1:bitcoin:xpub6DJBUQWdgF8c2afh1gY8T1679maU8nRxMQHKeoTxgkWfdWGnfpDnFzLR
        dWc5NghHk2VjvLTYts4Wb9PBP9m6t8LmkrdMn8rfD5L5n6iocK5
.....Public key:
        libcore:1:bitcoin:xpub6CUQpry1t11Dn1Q9D4HwCzjpgMdQzwR7MzvVe6kvvMFAj93RiAonDaFkYvE
        UNJppmG9dLqGQFWWzpVg9u4RdZMr9vCBcrAb3KLuVGKLQ73k
.....Public key:
        libcore:1:ethereum:xpub6BemYiVNp19ZzZoFuD8wsVuMyZD7tBPYuJFAcNZbKyJ49aHSGAHmSs
        D47ZzKyXF6SC91qaVxM4KxXYHVmDd5nyzadCpVW3a42r7tR1YqC4f
**********************************************
Account:
        libcore:1:bitcoin:xpub6DJBUQWdgF8c2afh1gY8T1679maU8nRxMQHKeoTxgkWfdWGnfpDnFzLR
        dWc5NghHk2VjvLTYts4Wb9PBP9m6t8LmkrdMn8rfD5L5n6iocK5
currencyId: bitcoin
satoshis: 1000
operations:
----------
hash: 2584924cf9f29f558966086da52d803d33e3b058cf7bf9d57b678c2b93855fff
.....date: 2019-12-02T18:47:14.000Z
.....satoshis: 1000
.....type: IN
.....senders: bc1qem70k77694v6grx8fe4ama9ju6xe8x0ylcnz7e
.....recipients: 345eNsfVgzvsK9HXd9ZHcky6WWZnYNHd4s bc1q3wxl6kq7vnn30e7ns8qr897ldhssky42s0vml6
```

University of New Haven

Calculating Artifact Integrity

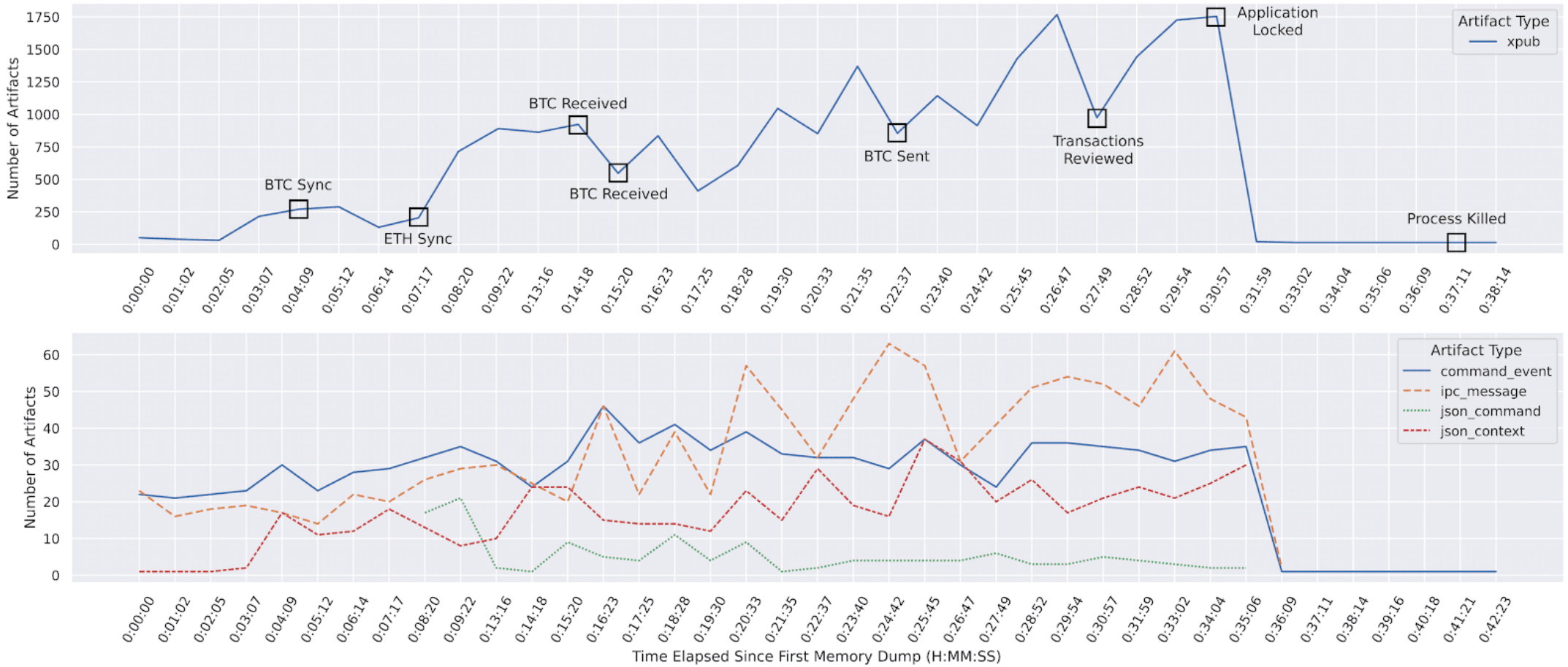
Determine:

- Are artifacts intact?
- How long do they stay intact?
- Are they usable by investigators?

Algorithm 2 IntegrityCalculator: Calculating Corruption of Memory Artifacts

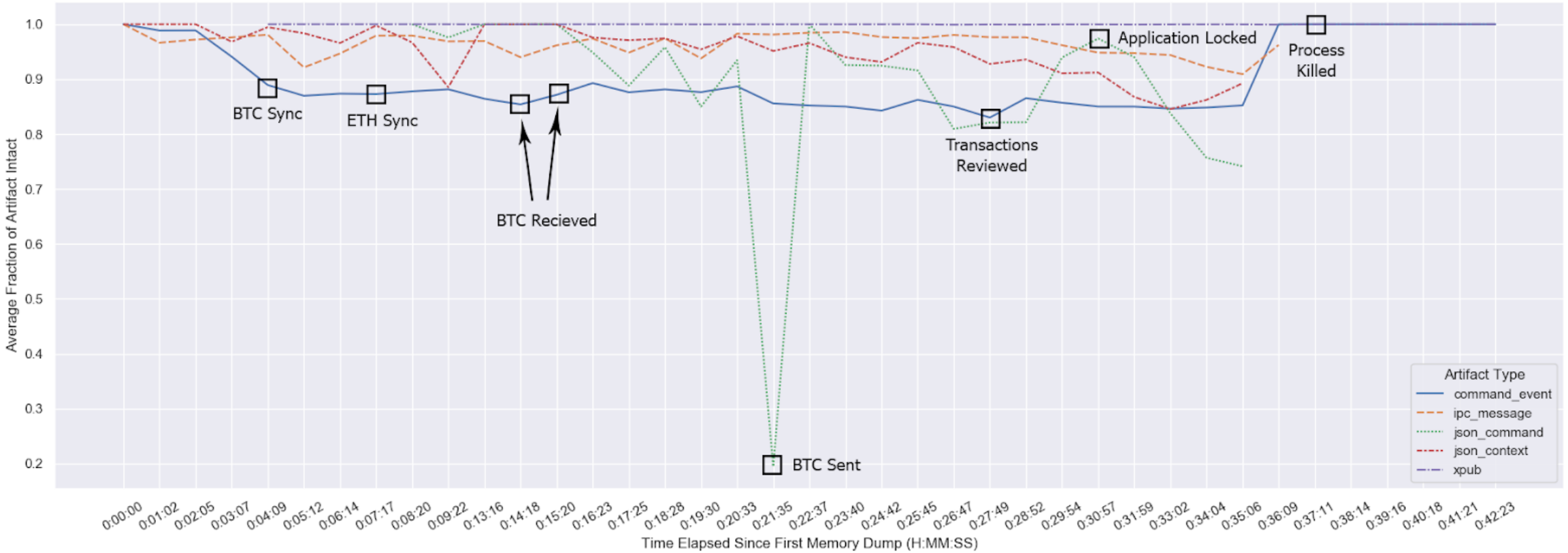
```
1:  function CalcMemDumpIntegrity(currData, prevData)
2:      for currArtifact ∈ currData do          ▷ Each artifact occurrence in memory dump
3:          currArtifact.integrity ← CalcArtifactIntegrity(currArtifact, prevData)
4:      end for
5:  end function
6:
7:  function CalcArtifactIntegrity(currArtifact, prevData)
8:      if containsArtifact(currArtifact, prevData) then
9:          prevArtifact ← matchArtifact(currArtifact, prevData)      ▷ Has occurred before
10:     else
11:         prevArtifact ← currArtifact                               ▷ First occurrence
12:     end if
13:     changeVector ← bytewiseAnd(currArtifact.data, prevArtifact.data)
14:     integrityVector ← bitwiseAnd(changeVector, prevArtifact.integrity)
15:     return integrityVector
16: end function
```

Ledger: Artifacts in Memory Over Time



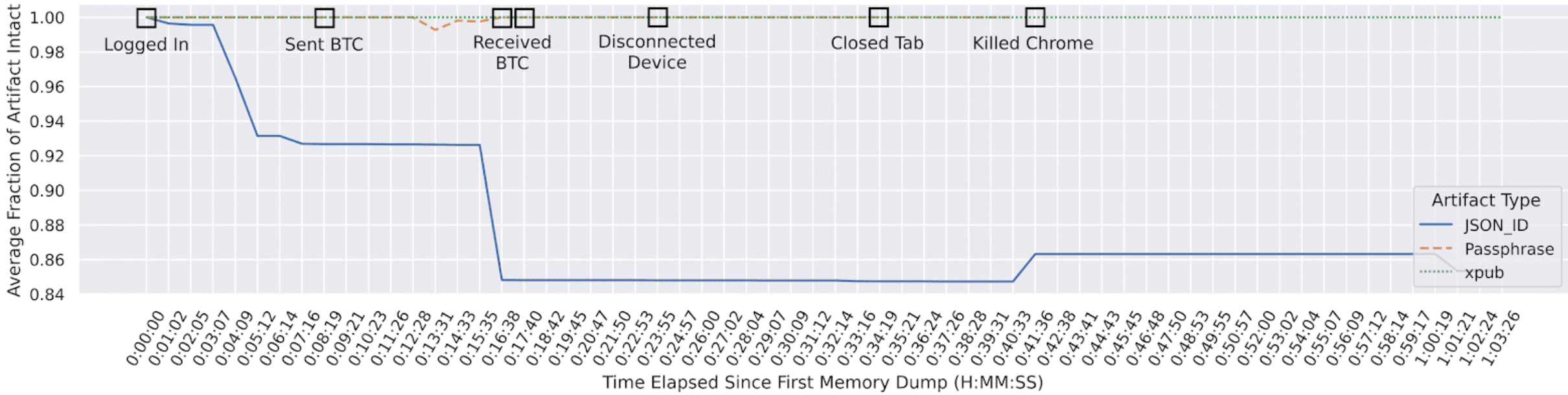
University *of* New Haven

Ledger: Artifact Integrity Over Time

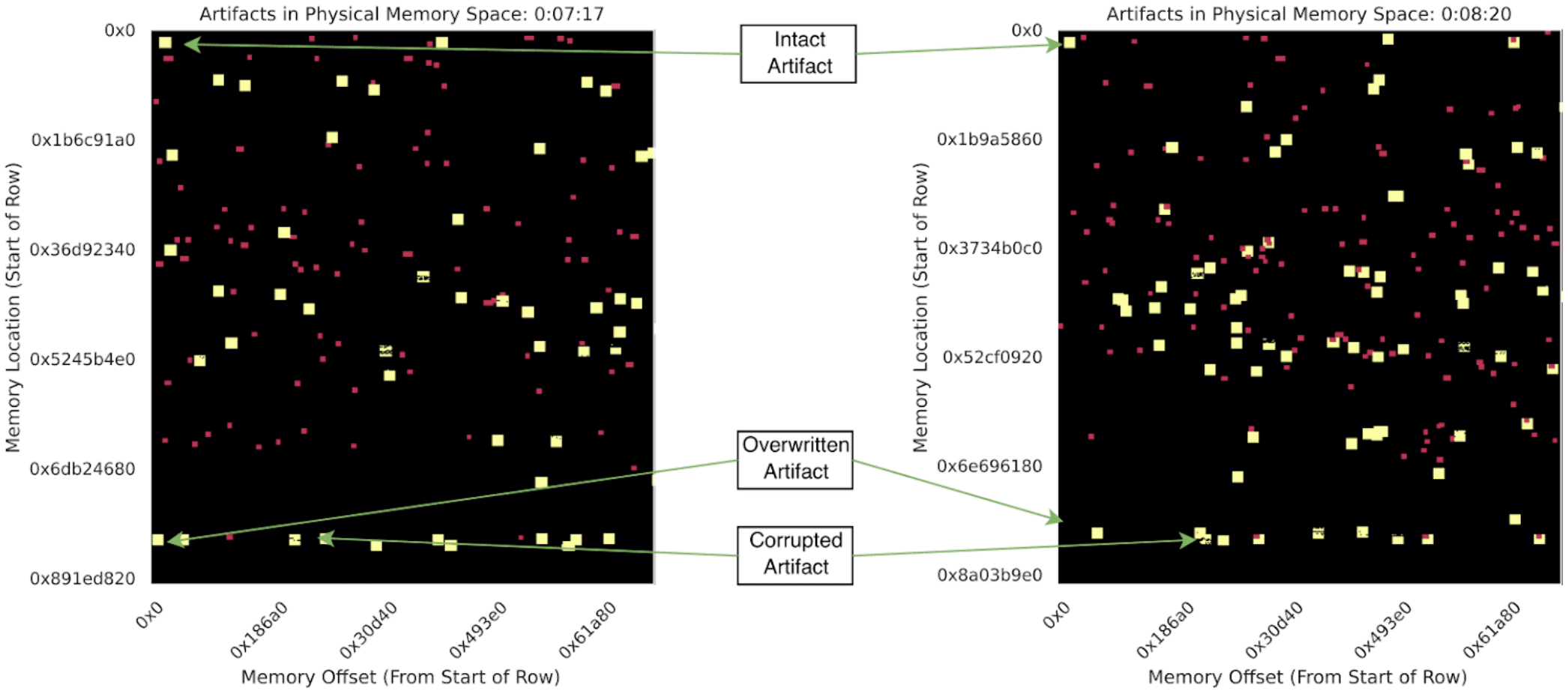


University *of* New Haven

Trezor: Artifact Integrity Over Time



Visualizing Memory Artifacts



University *of* New Haven

Discussion

- Obtaining an extended public key allows the derivation of all past and future wallet addresses
- This could be used to trace a transaction to an individual
- These keys are present in memory when the Ledger Live or Trezor Wallet clients are used
  - In some cases, transaction history and passphrases are also present
- A signficant amount of artifacts remain present after the browser tab is killed or the application is locked

UNHcFREG

Future Work

- Generalize object extraction across relevant technologies
- Lower level extraction allows for circumventing signature detection for each artifact
- Must interact directly with the run-time memory management system itself rather than signature-based detection

Acknowledgements

- National Science Foundation (NSF)
- University of New Haven
- The Volatility Foundation
- Ilya Shavrov
- Ibrahim Baggili
- UNHcFREG

UNHcFREG

Questions?

University *of* New Haven

UNHcFREG

References

1. Ali, S. S., El Ashmawy, A. and Shosha, A. F. (2018), Memory forensics methodology for investigating cryptocurrency protocols, in 'Proceedings of the International Conference on Security and Management (SAM)', The Steering Committee of The World Congress in Computer Science, Computer . . . , pp. 153–159.
2. BitcoinNews (2018), 'Hardware wallet sales booming as nano s tops1.3 million units'. URL: https://bitcoinnews.com/hardware-wallet-sales-booming-as-nano-s-tops-1-3-million-units
3. Case, A., Marziale, L. and Richard, G. G. (2010), 'Dynamic recreation of kernel data structures for live forensics', Digital Investigation 7, S32 – S40. The Proceedings of the Tenth Annual DFRWS Conference.
4. Case, A. and Richard, G. G. (2016), 'Detecting objective-c malware through memory forensics', Digital Investigation 18, S3 – S10. URL: https://doi.org/10.1016/j.diin.2016.04.01
5. Casey, P., Lindsay-Decusati, R., Baggili, I. and Breitinger, F. (2019),'Inception: Virtual space in memory space in real space'.
6. Garfinkel, S., Nelson, A. J. and Young, J. (2012), 'A general strategy for differential forensic analysis', Digital Investigation 9, S50 –S59. The Proceedings of the Twelfth Annual DFRWS Conference. URL: https://doi.org/10.1016/j.diin.2012.05.00
7. Gurkok, C. (2015), 'bitcoin.py'. URL: https://github.com/volatilityfoundation/community/blob/b4d65bd01870299c8c08e1e11b7b70dfe96cd1dd/CemGurkok/bitcoin.py
8. Ledger (2019), 'Celebrating 5 years at ledger then & now', Ledger. URL:https://www.ledger.com/celebrating-5-years-at-ledger-then-now/
9. Van Der Horst, L., Choo, K.-K. R. and Le-Khac, N.-A. (2017), 'Process memory investigation of the bitcoin clients electrum and bitcoin core'.
10. Zollner, S., Choo, K.-K. R. and Le-Khac, N.-A. (2019), 'An automated live forensic and postmortem analysis tool for bitcoin on windows systems', IEEE Access 7, 158250–158263