Computer Networks 273 (2025) 111691

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Survey paper

SoK: Design, vulnerabilities, and security measures of cryptocurrency wallets

Yimika Erinle [a,b], Yathin Kethepalli [c], Yebo Feng [d,*], Jiahua Xu [a,b]

[a] DLT Science Foundation, WC2H 2JQ, London, United Kingdom
[b] Centre for Blockchain Technologies, University College London, WCIE 6BT, London, United Kingdom
[c] GlueX Protocol, 2595VG, The Hague, Netherlands
[d] Nanyang Technological University, 639798, Singapore

ARTICLE INFO

Dataset link: https://github.com/xujiahuayz/crypto-wallets

Keywords:
Cryptocurrency wallet
Attacks
Defences
Key management
Wallet security
Wallet design

ABSTRACT

With the advent of decentralised digital currencies powered by blockchain technology, a new era of peer-to-peer transactions has commenced. The rapid growth of the cryptocurrency economy has led to the increased use of transaction-enabling wallets, making them a focal point for security risks. As the frequency of wallet-related incidents rises, there is a critical need for a systematic approach to measure and evaluate these attacks, drawing lessons from past incidents to enhance wallet security.

In response, we introduce a multi-dimensional design taxonomy for legacy and emerging wallets. We classify existing industry wallets based on this taxonomy, identify previously occurring vulnerabilities and discuss the security implications of design decisions. We also systematise threats to the wallet mechanism and analyse the adversary's goals, capabilities and required knowledge. We present a multi-layered attack framework and investigate 85 incidents between 2012 and 2025, accounting for a total loss of $6.98B. Following this, we classify defence implementations for these attacks on the precautionary and remedial axes. We map the mechanism and design decisions to vulnerabilities, attacks, and possible defence methods to discuss various insights.

1. Introduction

Pioneered by Bitcoin [1], peer-to-peer transactions have evolved into a digital ecosystem of decentralised financial applications on the blockchain. Building on this foundation with self-executing smart contracts on blockchain networks such as Ethereum, decentralised finance (DeFi) protocols enable decentralised lending [2], exchanges [3], derivatives [4], insurance [5], and numerous other financial applications [6–8]. As the user-facing component, wallets intricately trigger various transactions.

A wallet is a transaction-facilitating tool that manages user authentication to enable digital signing of transactions. It broadcasts these messages to a blockchain network to confirm their validity. When initiating a transaction, wallets use a private key to sign and broadcast the signature to the blockchain network [9]. Private key security is therefore critical, as incidents such as the Mt. Gox exchange attack (850,000 BTC) have resulted in significant financial losses for individual users and entities relying on the service [10]. Additional attacks on KuCoin [11], Vulcan Forged [12], Infarno [13], WazirX [14], and ByBit [15] have demonstrated that both custodial and non-custodial wallets present attractive targets.

This paper introduces a novel multi-dimensional cryptocurrency wallet taxonomy that extends beyond earlier approaches by covering both legacy and emerging wallets. The taxonomy reveals how specific design decisions correlate with known threat occurrences (Section 5). We systematise threats (Section 6) and attacks (Section 7), which enables us to suggest potential defence strategies (Section 8). We then discuss our analysis of design elements, attack vectors, and defence types in Section 10. In summary, our contributions are as follows:

- **Wallet Design Taxonomy:** We provide a taxonomy to analyse the design of various existing wallet types and propose new wallet designs. We also outline the threats to existing wallet designs based on our threat model.
- **Wallet Attacks Framework:** We systematise and analyse various attack methods, techniques and targets in literature. We then analyse 85 notable wallet incidents between 2012 and 2025 and investigate the attack gaps between academia and industry.
- **Defence Strategies:** We recommend defence methods based on the overall mitigation approach, incorporating both proactive and reactive approaches. We also analyse the influence of defence methods in mitigating attacks.

* Corresponding author.
E-mail address: yebo.feng@ntu.edu.sg (Y. Feng).

https://doi.org/10.1016/j.comnet.2025.111691
Received 27 April 2025; Received in revised form 14 August 2025; Accepted 4 September 2025
Available online 19 September 2025

To facilitate independent verification, all datasets and code used in this study are publicly available.[1]

2. Related works

2.1. Key management

Several studies have explored key management mechanisms. Courtois and Mercer [16] compare key management solutions with a focus on stealth addresses. Mangipudi et al. [17] investigate key management from the wallet users' perspective. He et al. [18] propose a secure key management scheme based on semi-trusted social networks. Di Angelo and Salzer [19] analyse the functionality of smart contracts for key management through transaction data. Most recently, Chatzigiannis et al. [20] propose a framework that formally evaluates hybrid recovery setups, highlighting key-management choices. Our study adopts a threat-centric view, mapping each key management technique in our multi-dimensional design taxonomy to specific attacks.

2.2. Wallet taxonomy

Prior research has proposed various methods to classify key management mechanisms [21–24]. Early wallet taxonomies by Bonneau et al. [21] and Eskandari et al. [22] survey key management techniques such as password-protected files, paper-based methods, hardware security module (HSM) systems, password-derived wallets, and hosted services. However, this classification was confined to a single axis of key storage. Karantias [23] contributes a protocol-centric taxonomy, examining light, full, and superlight clients and evaluating performance and security trade-offs. However, this approach does not extend to design elements such as key recovery methods or smart contract wallets, and lacks a mapping to threats and attack methods. Homoliak et al. [24] introduce an authentication-focused classification, examining k-factor and threshold-based co-signing solutions. While this emphasises the importance of multi-factor authentication in wallets, it only examines one of the several design elements we analyse.

By contrast, our taxonomy unifies multiple design dimensions into one integrative framework. These dimensions include custody model, key distribution, infrastructure (software or hardware), authentication, authorisation policies, and user recovery mechanisms. We include hardware wallets, exchange-based custodial solutions, shared-custodial implementations, non-custodial wallets, multi-party computation (MPC) wallets, and smart contract wallets in a consistent scheme. This approach bridges the gap between academic and industry viewpoints.

2.3. Wallet attack and security

A broad line of work surveys blockchain vulnerabilities and defences [25–28]. Chen et al. [27] focus on Ethereum's protocol-layer issues. Researchers also analyse specific wallet mechanisms; in particular, HSM-focused defence studies [29,30]. Additional studies investigate specific vectors such as phishing [31] and desktop-wallet RPC pitfalls [32]. Others scope security across wallet types [33], access key management impacts [34], and review attacks and defences in academia [35].

Our work differs by adopting a multi-layered defence perspective and incorporating real-world incident analysis to evaluate how design choices influence attacks. This approach bridges academic models with industry practice.

[1] GitHub repository at https://github.com/xujiahuayz/crypto-wallets.

2.4. Addressing literature gaps

Despite various studies on specific wallet types, mechanisms, and attack vectors, there is a lack of comprehensive examination spanning wallet design taxonomy, attack methods, incident analysis, security measures, and case studies, as shown in Table 1. Moreover, our design taxonomy is mapped with a detailed threat model and defence strategies, allowing a systematic evaluation of each design's security trade-offs. This comprehensive coverage and empirical attack data distinguish our work from prior classification-focused surveys. Our study bridges this gap, providing a holistic understanding crucial for advancing wallet security.

3. Generalised wallet mechanism

Cryptocurrency wallets facilitate state transitions by securely managing cryptographic keys and authorising transaction execution on the blockchain. To analyse wallet design and security, we first define a wallet. This definition underpins our mechanism, taxonomy, threat model, attack taxonomy, and security measures.

Definition 3.1 (*Cryptocurrency Wallet*)**.** A wallet is a system that typically generates a private key, also known as the secret key (*sk*), and securely stores it in an encrypted form (*enc_sk*), enabling an authenticated owner to sign transactions that are broadcast to the blockchain.

3.1. Key generation

The wallet initialisation process, detailed in Algorithm 1, specifies private key generation, public key generation, public address derivation and private key encryption for secure storage. As shown in Fig. 1, the internal flow of the wallet begins with private key (*sk*) generation from a random seed (*rdm_seed*). The corresponding public key (*pk*) is then derived from *sk* using the signature scheme and curve required by the target chain. Bitcoin, Ethereum, and Avalanche[2] all rely on the *Elliptic Curve Digital Signature Algorithm (ECDSA)* over the *secp256k1* curve by default [48]. Solana and Hedera default to the *Edwards-curve Digital Signature Algorithm (EdDSA)* curve *ed25519* [49], whereas the XRP Ledger supports both *ECDSA/secp256k1* and *EdDSA/ed25519*.

Once the key pair is generated and *pk* is obtained, the wallet hashes *pk* to produce the address (*addr*). Users share this address to receive funds. In account-based blockchains, the wallet queries *addr* via an remote procedure call (RPC) to fetch the current nonce (*nonce*). The nonce is initialised to 0 and preserves the sequential order of outgoing transactions.

Beyond curve selection, contemporary wallet software adheres to a concise suite of public standards. Bitcoin Improvement Proposal (BIP) 32 [50] and SatoshiLabs Improvement Proposal (SLIP) 10 [51] define hierarchical deterministic (HD) key derivation for *secp256k1* and *ed25519* curves, respectively.

Mnemonic phrases, as defined in BIP-39 [52], are the widely adopted standard for representing seeds in a human-readable form. SLIP-39 [53] extends this by applying Shamir's Secret Sharing to mnemonic phrases, enabling distributed or threshold-based recovery of wallet seeds. These mechanisms enable secure *sk* recovery in case of device loss or failure (see Section 5.8). At the account level, wallets use standard derivation paths such as BIP-44 [54], BIP-49 [55] and BIP-84 [56] to deterministically derive multiple accounts and address types from a single seed.

[2] Avalanche's C–Chain is Ethereum Virtual Machine (EVM) compatible and therefore inherits *secp256k1*. Hedera introduced optional *secp256k1* accounts in 2023 for EVM compatibility; however, *ed25519* remains the default.

Table 1
Overview of related works. (●: include, ○: not include).

Reference	Subjects covered						Methodology				Scope			
	Key cryptography	Key management	Key recovery	Attack methods	Security measures	Privacy techniques	Literature	Taxonomisation	Analysis	Case study	Wallet software	Wallet hardware	Smart contract wallet	Blockchain network
This study	●	●	●	●	●	○	●	●	●	●	●	●	●	○
[21]	●	●	○	○	●	●	●	●	●	○	●	○	○	●
[22]	○	●	●	○	●	○	●	●	●	○	●	○	○	○
[23]	○	○	○	○	●	●	●	●	●	○	●	○	○	○
[36]	○	●	○	●	●	○	●	●	●	○	●	●	○	●
[35]	○	●	●	●	●	●	●	●	●	○	●	●	○	●
[37]	●	●	●	○	○	○	●	●	○	○	●	○	○	○
[32]	○	○	○	●	●	○	●	●	●	○	●	○	○	○
[38]	○	○	○	○	●	●	●	●	●	○	●	○	○	●
[39]	○	○	○	●	●	○	●	○	●	○	●	○	○	○
[40]	●	●	○	●	●	○	○	●	●	○	●	○	○	○
[41]	○	○	○	●	●	○	●	○	●	○	●	○	○	○
[42]	●	●	●	●	●	●	●	○	●	○	○	●	○	○
[43]	●	●	●	○	●	○	●	●	●	○	○	●	○	○
[44]	○	○	○	●	●	○	●	●	○	○	○	●	○	○
[45]	○	○	●	○	○	●	●	●	○	○	○	●	○	○
[46]	●	●	●	○	●	○	●	○	●	○	○	●	○	○
[19]	○	○	○	○	○	○	●	●	●	○	○	○	●	○
[24]	○	●	○	○	●	○	●	●	●	○	●	●	○	○
[31]	○	○	○	●	●	○	●	○	●	○	○	○	○	○
[27]	○	○	○	●	●	○	●	●	●	○	○	○	○	●
[16]	●	●	○	●	●	●	●	○	●	○	●	○	○	●
[33]	●	●	○	○	●	○	○	○	●	○	●	○	○	○
[34]	○	●	○	○	●	○	○	●	●	○	●	●	○	○
[30]	●	●	○	●	●	○	○	○	●	○	○	●	○	○
[26]	○	●	○	●	●	○	●	●	●	○	●	●	○	●
[18]	○	●	●	○	●	○	○	○	●	○	●	○	○	○
[35]	●	●	●	●	●	○	●	●	●	○	●	●	○	●
[25]	○	●	○	●	●	○	○	○	●	○	●	○	○	○
[17]	○	●	○	○	○	○	○	○	●	●	●	○	○	○
[29]	●	●	○	●	●	○	○	○	●	●	○	●	○	○
[28]	○	○	○	●	●	○	●	●	●	○	○	○	●	●
[47]	○	●	●	●	●	○	○	○	●	●	●	○	●	○

Algorithm 1 Wallet initialisation

Input: *rdm_seed*: bin, *pw*: str
$sk = keyGen(rdm\_seed)$
$pk = publicKeyGen(sk)$
$enc\_sk = encrypt(sk, pw)$
$addr = hash(pk)$
$nonce = 0$

3.2. Key storage

Following its generation, *sk* is stored and encrypted using a key encryption key (KEK) that we refer to simply as the password (*pw*), as shown in Algorithm 1. In practice, *pw* is an abstract input that may be a traditional text password, a numeric PIN, a device-derived biometric secret, or a composite value obtained through multi-factor authentication (MFA). The ensuing key derivation function (KDF) output serves as the KEK. KEKs are typically derived with a password-based key derivation function (PBKDF), such as *PBKDF2-HMAC-SHA-256* [57], *scrypt* [58], or the memory-hard *Argon2id* [59]. The resulting KEK then protects *sk* under an authenticated encryption with associated data (AEAD) cipher such as *AES-256-GCM* [60] or *XChaCha20-Poly1305* [61]. The encrypted private key (*enc_sk*) remains secure, with *pw* required for both decryption and transaction signing. Secure *sk* storage is governed by the interplay of several factors described in Section 5.

3.3. Transaction management

Definition 3.2 (*Transaction*)**.** A transaction (*txn*) is a structured message created by a wallet that enables state change executions on the blockchain. These state changes include token transfers and smart contract interactions.

3.3.1. Transaction generation

Transaction generation begins with creating the transaction message (*txn*) by inputting the state transition information (*state_trans_info*). The message (*txn*) is then hashed to produce the transaction hash (*txn_hash*). Following transaction creation, the sender signs the transaction and provides *pw* to decrypt the private key (*sk*). The signing algorithm takes the decrypted private key (*sk*) and *txn_hash* as inputs to generate the signature (σ), which authorises the transaction (see Algorithm 2).

3.3.2. Transaction broadcast

The signature (σ) is verified using the sender's public key (*pk*) to assert its validity, as shown in Algorithm 3. If σ is invalid, the transaction is rejected and not processed further. Conversely, if σ is valid, the transaction is broadcast to the blockchain.

4. Methodology

Our methodology systematically bridges academic research and industry practice by analysing cryptocurrency wallet security across four axes: design, vulnerabilities, attacks, and defence measures.

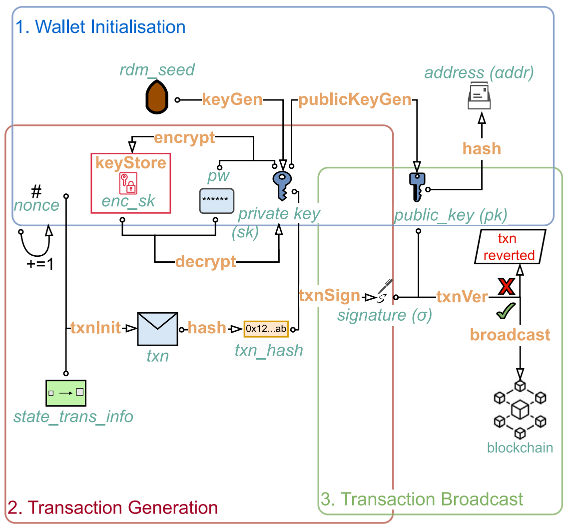

Fig. 1. Generalised cryptocurrency wallet mechanism showing Algorithms 1, 2 and 3.

4.1. Procedure

4.1.1. Design taxonomy and vulnerability

Our wallet design survey is structured as follows. We first perform reverse-engineering of specific vulnerable wallets to map vulnerabilities explicitly to underlying design features. Unique wallet features relevant to security were carefully documented and compared across wallet categories. The results of this analysis are summarised systematically in our wallet taxonomy table (Table 2), enabling structured comparisons and insight into security-usability trade-offs.

4.1.2. Attack methods

Following our design and threat analysis, we examine wallet attack methods in both academia and industry through a three-phase process. First, we conduct a comprehensive review of academic literature and industry incidents. We examine 33 peer-reviewed papers alongside 85 real-world incidents (2012–2025) documented in grey literature sources such as Rekt News and Slowmist.

To expand the reviewed literature scope, we conduct forward and backward reference searches. Following this, we categorise attacks using a three-tier framework to establish clarity and consistency. Attacks are classified hierarchically by their mechanism-centric goal (e.g. bypass the authentication mechanism), method (e.g., credential cracking), and vector (e.g., dictionary attack). We analyse industry incidents and identify patterns related to our design taxonomy or attack categorisation. Lastly, we perform a gap analysis to evaluate the alignment between academic research and industry practices.

Algorithm 2 Transaction generation

Input: *nonce*: int, *state_trans_info* : str, *enc_sk*: bytes, *pw*: str
Output: σ: bytes
nonce += 1
txn = *txnInit*(*state_trans_info*, *nonce*)
txn_hash = *hash*(*txn*)
sk = *decrypt*(*enc_sk*, *pw*)
σ = *txnSign*(*txn_hash*, *sk*)
return: σ

4.1.3. Security measures

Our security measures analysis begins by identifying proposed and implemented defensive strategies documented within the 33 academic papers focused on attack methods. We employ forward and backward reference searches to expand the scope of our reviewed literature to 61 unique references, retrieving an additional 28 academic papers. In addition, we consult grey literature sources on security measures. Each security measure is mapped to an identified wallet attack vector and classified based on the approach (e.g. proactive or reactive).

4.1.4. Case studies

We conduct in-depth case studies to illustrate the practical application of our framework. We systematically select representative wallet incidents based on severity and distinctiveness. Each case study follows a structured approach: (1) describing the wallet's design using our taxonomy, (2) detailing exploited vulnerabilities and threats, (3) outlining the adversary's goals, capabilities, and attack sequences and (4) recommending security measures. By integrating these real-world examples, we provide actionable insights into the interplay of wallet design, threats, and mitigation strategies.

4.2. Data sources

We sourced design variation, vulnerability, attacks and defence methods data from the following:

- **CVE Database:** We query the Common Vulnerabilities and Exposures (CVE) databases to retrieve previously identified wallet vulnerabilities.
- **Academic Papers:** We systematically retrieve academic papers, which serve as the primary data source for a range of wallet attack vectors and defence implementations.
- **Grey Literature:** We discover incidents on custodial and non-custodial wallets between 2012 and 2025, with most sources from Rekt News and Slowmist. Grey literature is also employed to retrieve additional vulnerabilities and security measures.

4.3. Inclusion criteria

Our resulting data conformed to the inclusion criteria below:

- **General Scope:** We limit our scope to exclude attacks on the blockchain protocol and on DeFi protocols from our discussion or analysis.
- **Vulnerability Inclusion:** We include wallet solutions with at least one CVE or previously detected vulnerability from searches.
- **Design Inclusion:** We include wallets with previously identified vulnerabilities, as well as those with significant user bases (such as MetaMask, Trust Wallet) or assets under management (AUM) (centralised exchanges such as Coinbase Exchange, Binance Exchange) and wallets with novel features (Argent, Safe (previously Gnosis Safe), ZenGo and Ngrave).
- **Attack and Defence Inclusion:** We include only attack methods and defence implementations, which can be mapped to key components within the underlying mechanism.
- **Case Studies Inclusion:** We include two notable incidents exhibiting: one custodial breach with the largest recorded monetary loss and one non-custodial compromise affecting the widest user base. These also provide comparative coverage of attacks against smart contract, hardware, and mobile wallet infrastructures.

Algorithm 3 Transaction broadcast

Input: σ: bytes, *pk*: hex
verified = *txnVer*(σ, *pk*)
assert(*verified*, "transaction failed")
broadcast(σ, *pk*)

Table 2
Industry wallet design variations and breadth of identified threat exposures, showing (for each wallet) the number and percentage of distinct threat categories observed (# and %). The denominator is the total number of threat types catalogued 15; a higher percentage means the wallet has experienced a greater diversity of threat categories. (●: include, ◐: part-inclusion, ○: not include) [62,63].

Name	Est.	Custody			Infrastructure								Initial.				Dis.			Authoris.			Valid.			Authentication					Recovery				Trans.		Chains Supported									Threat Occurrences															
					Software				Hardware																																																			
		Non-custodial	Shared-custodial	Custodial	Desktop	Browser	Mobile	Smart	USB	Bluetooth	NFC	QR code or air-gapped	Non-deterministic	Deterministic (Non-Hierarchical)	Hierarchical deterministic (HD)	Account contract	Single distributed	Multi-Sig	MPC	Single SK	Multiple SK	Relayer	Single PK validation	Multiple PK validation	Contract validation	PW/PIN	2FA	U2F	Passkey	Biometric	12W Seed	24W Seed	Social	DeRec	Open-source	Closed-source	Bitcoin (BTC)	Ethereum (ETH)	Polygon (MATIC)	BNB Chain (BNB)	XRP Ledger (XRP)	Hedera (HBAR)	Solana (SOL)	Cardano (ADA)	Avalanche (AVAX)	Inadequate encryption [62,64]	Insecure network [65–67]	Library vulnerability [68,69]	Insecure permission [70,71]	Predictable RNG [72,73]	Sig. Verif. logic flaw [74–77]	Side-channel leakage [78–80]	Data remanence [81,82]	Data manipulation [81,82]	Insecure interactions [83,84]	Inadequate authentication [63]	Input validation logic flaw [85]	Recovery Logic Flaw [86]	Provider compromise [87]	Insider compromise [69]	Threat #
Bitcoin Core	2009	●	○	○	●	○	○	○	○	○	○	○	●	○	●	○	●	○	○	●	○	○	●	○	○	●	○	○	○	○	○	○	○	○	●	○	●	○	○	○	○	○	○	○	○	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	3
Electrum	2011	●	○	○	●	○	○	○	○	○	○	○	●	○	●	○	●	●	○	●	●	○	●	●	○	●	●	○	○	○	●	○	○	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	1
Coinbase Exchange	2012	○	○	●	○	◐	◐	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●	●	○	●	●	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	0
Trezor	2013	●	○	○	○	○	○	○	●	○	○	○	○	○	●	○	●	●	○	●	●	○	●	●	○	●	○	●	○	○	●	●	○	○	●	○	●	●	●	●	●	○	●	●	●	○	○	○	○	○	●	●	●	●	●	○	○	○	○	○	5
Kraken Exchange	2013	○	○	●	○	◐	◐	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●	●	○	●	○	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	0
Ledger	2014	●	○	○	○	○	○	○	●	●	○	○	○	○	●	○	●	○	○	●	○	○	●	○	○	●	○	●	○	○	○	●	○	○	◐	○	●	●	●	●	●	●	●	●	●	○	○	●	○	○	○	●	○	○	●	○	○	○	○	●	4
MetaMask	2016	●	○	○	○	●	●	○	○	○	○	○	○	○	●	○	●	○	○	●	○	○	●	○	○	●	○	○	○	●	●	○	○	○	●	○	○	●	●	●	○	●	○	○	●	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	1
Exodus	2016	●	○	○	●	●	●	○	○	○	○	○	○	○	●	○	●	○	●	●	○	○	●	○	○	●	○	○	●	●	●	○	○	○	○	●	●	●	●	●	●	●	●	●	●	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	1
Binance Exchange	2017	○	○	●	◐	◐	◐	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●	●	●	●	●	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	0
Trust Wallet	2017	●	○	○	○	●	●	○	○	○	○	○	○	○	●	○	●	○	◐	●	○	○	●	○	○	●	○	○	○	●	●	○	○	○	●	○	●	●	●	●	●	○	●	●	●	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	1
Argent	2017	●	○	○	○	●	●	●	○	○	○	○	○	●	○	●	○	●	○	○	●	●	○	○	●	○	○	○	●	○	○	○	●	○	●	○	○	●	●	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	●	○	○	2
Safe (Gnosis)	2017	●	○	○	○	○	●	●	○	○	○	○	○	●	○	●	○	●	○	○	●	●	○	○	●	○	○	○	●	○	○	○	●	○	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	●	○	○	○	○	2
Atomic	2017	●	○	○	●	○	●	○	○	○	○	○	○	○	○	●	●	○	○	●	○	○	●	○	○	●	○	○	○	○	●	○	○	○	○	●	●	●	●	●	●	●	●	●	●	○	○	○	○	●	●	○	○	○	○	○	○	○	○	○	2
Tangem	2017	●	○	○	○	○	○	○	○	○	●	○	○	○	●	○	●	○	○	●	○	○	●	○	○	●	○	○	○	●	●	●	○	○	●	○	●	●	○	●	●	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	0
Ngrave	2018	●	○	○	○	○	○	○	○	○	○	●	○	○	●	○	●	○	○	●	○	○	●	○	○	●	○	○	○	●	○	●	○	○	○	●	●	●	○	●	●	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	0
ZenGo	2018	○	●	○	○	○	●	○	○	○	○	○	○	●	○	●	○	○	●	●	○	○	●	○	○	○	●	○	○	●	○	○	○	○	●	○	●	●	●	●	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	1
Coinbase Wallet	2019	●	○	○	○	●	●	●	○	○	○	○	○	○	○	●	●	○	○	●	○	●	○	○	●	○	○	○	●	○	●	○	●	○	○	●	●	●	●	●	●	○	●	●	●	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	1
Biconomy	2019	●	○	○	○	○	○	●	○	○	○	○	○	○	○	●	●	○	○	●	○	●	○	○	●	○	○	○	●	○	○	○	●	○	●	○	○	●	●	●	○	○	○	○	●	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	1
Web3Auth	2020	○	●	○	○	○	●	○	○	○	○	○	○	●	○	●	○	○	●	○	○	●	○	○	●	○	○	●	●	○	○	○	●	○	●	○	○	●	●	●	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	1
Brave	2021	●	○	○	○	●	●	○	○	○	○	○	○	○	●	○	●	○	○	●	○	○	●	○	○	●	○	○	○	●	●	○	○	○	●	○	●	●	●	○	○	○	●	○	○	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	2
Phantom	2021	●	○	○	○	●	●	○	○	○	○	○	○	○	●	○	●	○	○	●	○	○	●	○	○	●	○	○	○	●	●	●	○	○	○	●	●	●	●	○	○	○	●	○	○	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	2
Slope	2021	●	○	○	○	●	●	○	○	○	○	○	○	○	●	○	●	○	○	●	○	○	●	○	○	●	○	○	○	●	●	○	○	○	●	○	○	●	○	●	○	○	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	●	○	2
HashPack	2021	●	○	○	○	●	●	○	○	○	○	○	○	○	●	○	●	○	○	●	○	○	●	○	○	●	○	○	○	●	●	○	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	1
Binance Web3 Wallet	2023	○	●	○	○	○	●	○	○	○	○	○	○	●	○	●	○	○	●	●	○	○	●	○	○	○	○	○	●	●	○	○	○	○	●	○	○	●	●	●	○	○	●	○	●	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	1
TON Space	2023	●	○	○	○	○	○	●	○	○	○	○	○	○	●	●	○	○	○	●	○	○	●	○	○	●	●	○	○	●	○	●	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	●	○	○	○	○	○	2
Kraken Wallet	2024	●	○	○	○	○	●	○	○	○	○	○	○	○	●	●	●	○	○	●	○	○	●	○	○	○	○	○	●	●	●	○	○	○	●	○	●	●	●	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	0
Summary					**Highest Occurrence: Signature verification logic flaw**															7																																							**Total threat instances across all wallets**	36(100%)

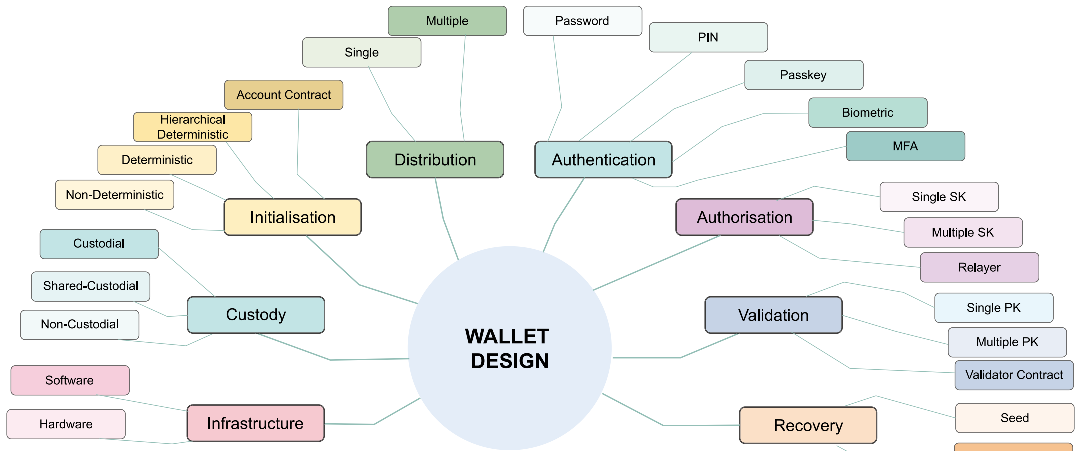

Fig. 2. Multi-dimensional wallet design taxonomy for traditional and emerging wallets. Fig. 3 maps industry wallets on three dimensions based on this taxonomy.

5. Wallet design taxonomy

We propose a design taxonomy for classifying and developing wallets that integrates traditional models and recent advances, as illustrated in Fig. 2. To develop this framework, we analyse various designs of wallets within the industry. We also identify known vulnerabilities and previous attacks associated with these wallets, as summarised in Table 2.

5.1. Infrastructure

This design factor is centred on the private key (*sk*) or transaction management infrastructure (see Section 3) the controlling entity employs.

5.1.1. Software wallets

Software wallets are applications that manage private keys (*sk*) or transaction authorisation conditions within a software environment. Existing software infrastructure designs include desktop, browser, mobile and smart contract wallets, as demonstrated within Fig. 3. Desktop wallets are installed on computers and typically store *enc_sk* in a local file within the computer's file system. Browser wallets present an alternative setup, with programs installed or built into the web browser and credentials are typically stored in the browser's local storage [88]. Two existing designs are browser extensions, such as MetaMask and Phantom, and built-in browser-native wallets, such as Brave [89].

Another prevalent wallet type is the mobile wallet, which is installed on devices with limited computing power and storage capability in comparison with PCs. Mobile wallets also typically store *enc_sk* locally and can enhance security through mobile OS integrations such as the Android Keystore and iOS Keychain [90]. However, if vulnerabilities are present in the operating system Section 6.1, susceptibility to specific attacks that exploit these weaknesses exists (see Section 7.2.3).

To mitigate the risk of *sk* and *rdm_seed* loss, smart contract wallets (e.g., Argent and Safe) are deployed on the blockchain to abstract typical *sk* management (see Section 3) and create advanced transaction functions such as multi-factor authentication (MFA), ownership assignments, spending limits, and recovery mechanisms, often through integration with centralised or decentralised relayers [19,91].

TON Space, another smart contract wallet, allows users to create and sign transactions without leaving the chat by interacting through TON's standard Wallet-V4 account model [92]. The key management functionality, bot-based transfers, and cloud backups are mediated through Telegram IDs and WebView sessions. This approach shifts part of the trust boundary from the mobile operating system to Telegram's API and bot infrastructure, introducing centralisation risks [93] and exposing generic WebView attack surfaces [94]. Despite their capabilities, smart contract wallets are susceptible to library vulnerabilities, implementation flaws, and access-control misconfigurations. These application logic vulnerabilities have resulted in significant financial losses in several cases [15,95,96].

5.1.2. Hardware wallets

Hardware wallets typically involve *sk* management within a secure element (SE) (e.g., microcontroller or smart card) to protect against tampering and facilitate the execution of cryptographic operations, such as transaction signing (see Section 3). Isolated in design with no internet connectivity, their mechanism performs all cryptographic operations on an offline hardware device. They typically require a distinct online device to create and broadcast transactions [97]. As shown in Fig. 3, the connection between both devices can be achieved through Bluetooth (e.g., Ledger), USB (e.g., Trezor), Near Field Communication (NFC) (e.g., Tangem) and QR codes (e.g., Ngrave). Specific hardware wallet vulnerabilities [98–101], and attacks [102–105] are discussed in Sections 6.1.4 and 7.4 respectively.

5.2. Custody

The degree of *sk* control by an entity or between one or more entities defines custody design. Custody setups include custodial, non-custodial and shared-custodial.

5.2.1. Custodial

In this model, *enc_sk* is stored by a trusted custodian (e.g., Coinbase Exchange, Binance Exchange, Kraken Exchange) who signs user-initiated transactions. The user relinquishes *sk* security to the custodian who fully controls the wallet operations (see Section 3), while the user solely crafts transaction messages. Although most of the design factors for custodial wallets are not disclosed (see Table 2), a classification of their design can be conducted using our framework. In the table, we denote "◐" representing user-facing infrastructure and "●" the internal infrastructure employed by the custodian.

Two notable design variations exist in custodial wallets. First, an omnibus setup aggregates and controls all users' funds under a few shared addresses, without a one-to-one correspondence between user accounts and addresses. Second, a segregated setup assigns each user a unique blockchain address, with the custodian retaining control of the associated private keys (*sk*) [106].

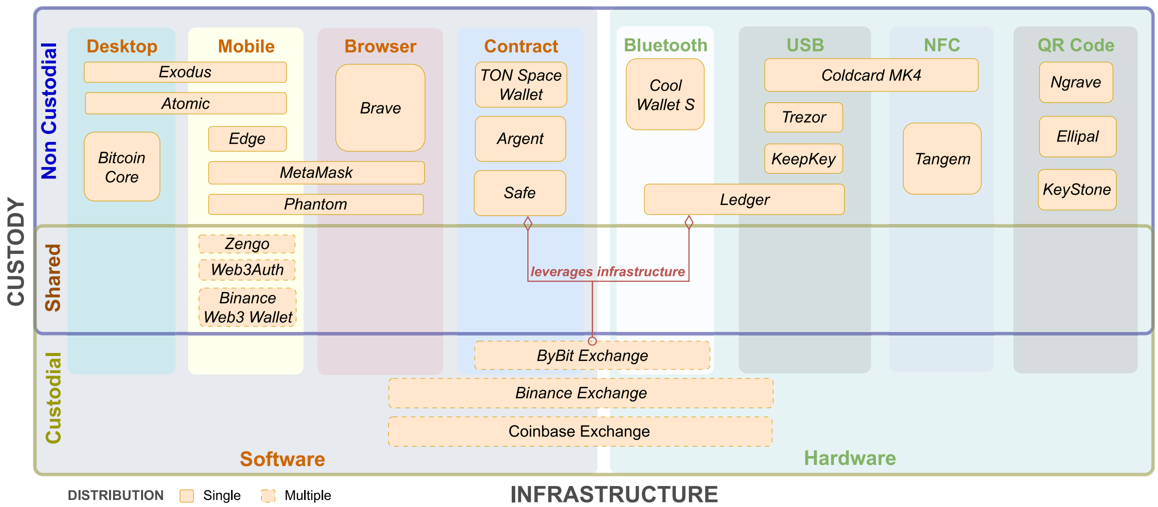

Fig. 3. Wallet design taxonomy showing three of our eight dimensions (infrastructure, custody, and distribution), as detailed in Sections 5.1, 5.2 and 5.4.

5.2.2. *Non-custodial*

In non-custodial wallet architectures (e.g., MetaMask, Phantom, Ledger), the user does not relinquish control to any custodian party. Instead, a direct interaction between the user and the blockchain network exists in these setups with the user in full control of *sk*, to facilitate all the wallet operations (see Section 3). With full autonomy, the user is solely responsible for securing *sk* and is more susceptible to insecure user interaction threats as well as other vulnerabilities (see Section 6.1) and attacks such as social engineering attacks and malware-based attacks (see Section 7.2) which aim to exploit user negligence. While non-custodial wallets are expected not to have credential control, a few incidents in the past (e.g., Slope Wallet [107]) have resulted in *sk* compromise due to poor implementation practices, insecure storage of sensitive information, or inadvertent leaks [87].

5.2.3. *Shared-custodial*

Shared-custodial wallets strike a balance between custodial and non-custodial models by enabling joint control of the secret key (*sk*) between a user and a custodian. In this setup, the *sk* is split or distributed across two or more parties, allowing the user to delegate a degree of transaction authorisation rights and trust to the custodian. This arrangement gives both parties partial control over the wallet's signing and recovery operations [108,109]. As a result, even if one party's security is compromised, the risk of a complete *sk* compromise is mitigated. For example, ZenGo's operational model implements shared custody with multi-party computation (MPC) by storing one part of the *sk* on ZenGo's centralised server, while the other part remains on the user's device [110]. Other shared custodian models are discussed in Section 5.4.

5.3. *Initialisation*

This pertains to the creation of the wallet through *sk* generation (see Section 3.1) or contract deployment. During initialisation in smart contract wallets, user account contracts are typically created by interactions made by the relayer. In conventional wallets, the *sk* generation scheme can be non-deterministic, deterministic, or hierarchical deterministic, depending on the degree of randomness and flexibility required. Another interesting design option is the KDF choice. Typically, most wallets (e.g., Ledger [111]) employ password-based key derivation function (PBKDF); however, novel research into threshold multi-factor key derivation function (MFKDF) construction could influence current cryptographic designs [112,113]. While this improves security, more processing time and power may be required to generate the derived key [81].

5.4. *Distribution*

This is the degree of authorisation (see Section 5.6) or *sk* distribution between storage mechanisms. Single or variations of shared authorisation between multiple user devices, multiple users or a user and a custodian (see Section 5.2) are observable setups. Single setups allow for sole authorisation by a user or custodian, while authorisation is distributed in the shared setup to avoid a single point of failure.

Multi-distributed designs typically exist in two forms: smart wallet-enabled multi-sig (on-chain multi-sig) and threshold MPC. For smart contract wallets that follow Ethereum Improvement Proposal (EIP) 4337, the account contract may adopt any of these schemes: single key, multi-sig, or MPC, as the standard merely asks the contract to prove validity to `validateUserOp`. On-chain multi-sig typically has authorisation dispersed between multiple private keys (*sk*), while MPC wallets divide a single *sk* into "key shares", which are then distributed [114,115]. Design flexibility in some MPC wallets also allows for a hierarchical sub-shard distribution (e.g., Web3Auth) if necessary [116]. While both offer authorisation distribution, trade-offs exist between the two (see Sections 5.6 & 5.7).

5.5. *Authentication*

Authentication is the process of verifying the legitimate wallet owner before granting access, either by decrypting *enc_sk* with the key encryption key (KEK) (see Section 3.2) or by employing other methods defined within the underlying logic. Existing authentication methods include single-factor (*pw* or *PIN*), multi-factor authentication, and novel password-abstracted authentication methods such as passkeys enabled by smart contract or MPC wallets. For instance, the Binance Web3 Wallet uses MPC to generate three key-shares: one secured by Binance, one stored on the user's device, and one encrypted with a user-defined recovery password and backed up to the user's iCloud/Google Drive. The wallet uses a 2-of–3 threshold scheme to authorise transactions, so Binance's single share is insufficient on its own [117].

5.6. *Authorisation*

Authorisation in the context of wallets is defined as a direct or indirect confirmation of a state change transaction (see Definition 3.2) by a single signature or multiple signatures (σ). In the EIP-4337 flow, the user signs a `UserOperation`. However, a Bundler/Relayer authorises the on-chain transaction by submitting the batch to the shared `EntryPoint` [91]. We therefore mark every 4337 wallet as "Relayer" in Table 2. MPC key shards produce a single signature while being distributed among various parties with individual public addresses hidden.

Multi-sig smart wallets demonstrate authorisation through multiple signatures, each associated with an individual public address. This approach does not enhance privacy since all involved addresses are visible on the blockchain. EIP-4337-enabled smart contract wallets employ a relayer (bundler) to aggregate multiple users' state transfer messages into a single authorised transition. Another factor that influences the authorisation setup is the choice of signature scheme.

5.7. Validation

Transaction validation typically refers to authentication against the blockchain using the user's *pk* [24,36]. In addition to single distributed wallets, MPC wallets also produce a single *pk* from key shards, which can be employed to validate the transaction. On the other hand, native multi-sig wallets validate each party's public key. EIP-4337 allows more flexible validation variations, as an `EntryPoint` contract validates and executes state changes sent by authenticated users [91]. Additionally, recent developments (ERC-1271 [118] & ERC-6492 [119]) have enabled standardised and improved signature validation methods for smart contracts.

5.8. Recovery

Recovery serves as a method to retrieve *sk* or lost transaction authorisation rights and typically follows the initialisation (see Section 5.3) and distribution (see Section 5.4) setups selected.

5.8.1. Seed recovery

The industry standard fallback for a user wallet is the Bitcoin Improvement Proposal (BIP) 39 mnemonic recovery phrase, usually 12 or 24 English words that encode the master hierarchical deterministic (HD) seed [52]. Specifically, 128–256 bits of random entropy are appended with a checksum and split into 11-bit chunks, each of which indexes one word in the 2048-word BIP-39 list. When the user re-enters the phrase, it is processed through 2048 iterations of PBKDF2 (Password-Based Key-Derivation Function v2) using HMAC-SHA-512 (a keyed-hash message-authentication code) to yield a 512-bit seed. This seed forms the root of the wallet's HD key tree, from which every subsequent *sk* is derived [52].

Two notable design variations to the default mnemonics setup exist to offer additional security. First, the optional portability passphrase ("25th word") in BIP-39 allows plausible deniability if the base phrase is coerced [120]. Second, SLIP-39 Shamir-Secret-Sharing mnemonics fragment the seed into shares, requiring a quorum (e.g., m-of-n) to restore the wallet [121,122]. Some mobile wallets go further by pairing mnemonics with encrypted cloud backups (e.g., Coinbase Wallet using iCloud/Google Drive), improving usability while keeping control with the user [123].

Despite convenience gains, mnemonic phrases remain prime targets for social engineering and clipboard-scraping malware, reinforcing the need for offline generation and, where feasible, distributed-share approaches. Social platforms such as Telegram extend cloud backups into custodial-assisted models. For example, TON Space encrypts the seed locally and synchronises it with Telegram Cloud, binding recovery to the user's Telegram ID. After re-authenticating that account, the Mini App reinjects the seed into a Wallet-V4 contract. This incurs no on-chain fee; however, it creates a single point of failure, as loss or compromise of the Telegram account threatens both availability and confidentiality [92,93].

5.8.2. Seedless recovery

Seedless recovery eliminates mnemonic phrases and re-establishes user authorisation rights without a seed. Single or multi-party variations exist, with common instantiations including contract-based social recovery, MPC re-sharing, and other implementations such as Decentralised Recovery (DeRec) [115,124,125]. Implementations differ and create distinct cost profiles in smart contract and MPC wallets. MPC wallets perform recovery off-chain through key fragment reconstruction and thus incur no on-chain network fees. By contrast, smart contract wallets (e.g., Coinbase Smart Wallet) implement recovery as an on-chain signer/owner change that requires a network fee [126]. However, one smart contract wallet, Argent, circumvents this by offering users off-chain recovery [127]. More recently, the DeRec standard proposes an interoperable, multi-party key recovery framework that allows users to regain access across different wallets and services without relying on a single custodian [125].

5.9. Other design factors

Table 2 shows other design factors such as transparency and agnosticism. The underlying mechanism of existing hardware, software, non-custodial and shared-custodial wallets often functions in degrees of transparency. While open-source models benefit from public audits, open knowledge of mechanisms can provide an advantage to an adversary. Chain support is another important factor, as integration with multiple blockchain networks defines blockchain-agnosticism. As blockchains often operate as fragmented systems, heterogeneous designs foster enhanced interoperability.

6. Threat model

We analyse threats to the wallet mechanism, considering adversary goals, knowledge, and capabilities. Using our design taxonomy (Table 2), we also identify industry threats and highlight gaps between industry and academia (Table 3).

6.1. Classification

Our threat classification is structured around distinct operations within the wallet mechanism across three stages: wallet initialisation, transaction generation, and transaction broadcast. Threats to the system can be categorised into five areas: network, authentication, application, storage and memory, and cryptanalysis.

6.1.1. Network

The wallet communicates with the blockchain to retrieve and broadcast *state_trans_info* using internet protocols. The network enables the secure transmission of messages within and outside of the system. Vulnerabilities in the communication channels can be targeted, as shown in Table 5. Service providers in the network can also be compromised, rendering messages vulnerable to interception and alteration.

6.1.2. Application

Wallets rely on application libraries [130], and operating systems [39,131], which may possess vulnerabilities that the adversary can exploit. Vulnerabilities in these systems include application logic vulnerabilities such as key recovery [86], signature verification [74], and input validation [85] flaws, which can result in privilege escalation. Additionally, malware exposure [39,138], insecure third-party interactions [83,84], and user negligence [139] can threaten the security of the *sk*, *rdm_seed*, or *pw*. For instance, projects integrating TON wallets have experienced silent exfiltration of mnemonic phrases via malicious application libraries [140,141]. Web3 wallets embedded in social platforms amplify supply-chain risk. Malicious npm modules impersonating TON SDKs (e.g., `@ton-wallet/create`) execute clipboard-sniffers that forward seed phrases to attacker-controlled Telegram bots [140,142]. Since wallet logic is tightly coupled to chatbot APIs, a single rogue Mini App link can invoke in-chat transaction authorisation by users, as seen in the June 2025 phishing wave [143].

Table 3
Classification of threats (Section 6.1) showing the targeted operation, and the capabilities (Section 6.3), knowledge and accessibility of the adversary. A gap analysis of threats is also conducted to compare industry and academia [64–73,78–80,128,129].

Category	Threat	Gap		Target						Adversary's capability summary	Knwl.			Acc.	
		Academia	Incidents	KeyGen	TxnInit	UserAuth	KeyStore	TxnSign	TxnVer		Public	Restricted	Insider	Remote	Physical
Network	Insecure Network Channel [65–67]	●	●	○	●	○	○	○	○	Intercept transactions or deny service to wallet.	●	○	○	●	○
	Compromised Network Protocol [130]	●	○	○	○	●	○	○	○	Misroute transactions via protocol flaws.	●	○	○	●	○
Application	Application Logic Flaw [96,129]	●	●	○	○	●	○	○	○	Exploit the programming logic of functions.	●	○	○	●	○
	OS Vulnerabilities [131]	●	●	○	○	○	●	○	○	Exploit OS to bypass security.	○	●	○	●	○
	Library Vulnerability [68,69]	●	●	●	○	○	○	●	●	Exploit vulnerabilities in third-party libraries.	●	●	○	●	○
	Coding Errors [96]	●	●	○	●	●	○	○	○	Exploit coding errors to bypass security.	●	○	○	●	○
	Insecure Interaction [83]	●	●	○	●	○	○	○	○	Exploit users through UI deception.	●	●	●	●	○
	Application Provider Compromise [132]	○	●	○	●	○	○	○	○	Exploit provider infrastructure to inject code.	●	○	○	●	○
	Data Misrepresentation [133]	○	●	○	●	○	○	○	○	Misrepresent transaction data on user interface.	○	●	●	●	○
Authentication	Inadeq. Authentication [134]	●	●	○	○	●	○	○	○	Bypass authentication mechanisms.	●	●	○	●	●
	Low-strength Password [41,135]	●	●	○	○	●	○	○	○	Brute-force weak passwords.	●	○	○	●	○
Storage	Insecure Boot Environment [136]	●	○	○	○	○	●	○	○	Hijack boot to execute malicious code.	○	●	○	○	●
	Insecure Permissions [70,71]	○	●	○	○	○	●	○	○	Escalate file-system permissions.	○	●	○	●	●
	Inadequate Encryption [64,87]	●	●	●	○	○	●	●	○	Read unencrypted secrets at rest.	○	●	●	●	●
	Data Remanence [81,82]	●	●	○	○	○	●	○	○	Recover keys from memory remnants.	○	●	○	○	●
	Data Manipulation [81,82]	●	●	○	○	○	●	○	○	Tamper with stored data.	○	●	○	●	●
	Micro-electrical Exposure [105]	●	●	○	○	○	●	○	○	Probe chip side-effects.	○	●	○	○	●
	Storage Provider Compromise [87]	○	●	○	○	○	●	○	○	Breach external storage vendor.	○	●	●	●	○
Cryptanalysis	Predictable RNG [72,73]	●	●	●	○	○	○	○	○	Predict or replay RNG outputs.	●	●	○	●	○
	Weak Signature [137]	●	●	○	○	○	○	●	●	Forge signatures under weak crypto.	●	●	○	●	○
	Side-channel Leakage [78–80]	●	●	○	○	○	●	○	○	Extract secrets via side-channels.	●	●	○	●	●
Other	Insider Collusion [128]	○	●	○	○	●	●	○	○	Collude with insiders.	○	○	●	●	●
	Insider Compromise [69]	○	●	○	○	●	●	○	○	Abuse insider access.	○	○	●	●	●

6.1.3. Authentication

Authentication is a critical process in modern wallets, as only an authorised owner can decrypt an encrypted private key (*enc_sk*) and sign transactions (refer to the *encrypt* and *decrypt* functions in Algorithm 1 and Algorithm 2, respectively). Authentication attacks aim to compromise the wallet function that verifies the user's identity, thereby gaining unauthorised access to wallets. The authentication functions, which handle the encryption and decryption of the *enc_sk*, can be vulnerable to insecure boot environments [136] and single-factor authentication methods and low-strength passwords (*pw*).

6.1.4. Storage and memory

Data stored can be vulnerable to threats of extraction, manipulation and disruption. Exploitation of the wallet's storage mechanism (see Section 3.2) can lead to the compromise of *sk*, *rdm_seed* or *pw*. Storage mechanism vulnerabilities include data remanence [136], unencrypted data [144,145], and physical security vulnerabilities [105] that can be exploited by the adversary.

6.1.5. Cryptanalysis

Cryptographic vulnerabilities may exist in the signature scheme (*keyGen*, *txnSign*, *txnVer*) as a result of the direct implementation or unintended data leakages from side channels. These vulnerabilities include hash function vulnerabilities [146], weak signatures (σ) [137], predictable random number generator (RNG) [147], and data leakages from side-channels [148,149].

6.1.6. Other threats

Threats can occur via other avenues, such as an insider who may have access to transactional information, user credentials and other security details. These can arise from insiders acting maliciously or by exploitation through coercion or social engineering methods. Custodial (Section 5.2.1) and Shared-custodial (Section 5.2.3) architectures are more vulnerable to these threats due to their more centralised architecture. Non-custodial setups (see Section 5.2.2) may be vulnerable if third-party services are employed for functionalities such as *pw* management or if inadequate access controls are relied upon (e.g., Ledger incident [150]).

6.2. Adversary's goals

We define an adversary, $\mathcal{A}$, who aims to exploit threats described above to trigger unauthorised transactions to an adversary-controlled wallet address or disrupt operations. The major goals of $\mathcal{A}$ include:

- **Credential Compromise:** $\mathcal{A}$ aims to compromise *sk*, *rdm_seed* and *pw* by exploiting wallet mechanism vulnerabilities or user-interactions.
- **State Transition Information Manipulation:** $\mathcal{A}$ aims to modify the *state_trans_info* created by the user such as *recipient_address*. Following this, $\mathcal{A}$ deceives the user into signing the transaction. $\mathcal{A}$ may also manipulate the *state_trans_info* displayed on the wallet interface

6.3. Adversary's capabilities

Table 3 details the various capabilities of $\mathcal{A}$, illustrating how identified vulnerabilities can be exploited to achieve an objective with various degrees of knowledge and access. $\mathcal{A}$ can possess public, restricted and insider knowledge. Public knowledge includes information that is openly accessible to anyone, such as open-source code, publicly available audit reports, discussions in open forums, websites, and applications. Restricted knowledge refers to information that is not readily accessible to the public and often requires specific roles, permissions, or effort to obtain. Information that is only accessible to individuals within an organisation is defined as insider knowledge, particularly in setups where custodians have some level of authorisation (Section 5.2). $\mathcal{A}$ can also execute several attack capabilities remotely or physically.

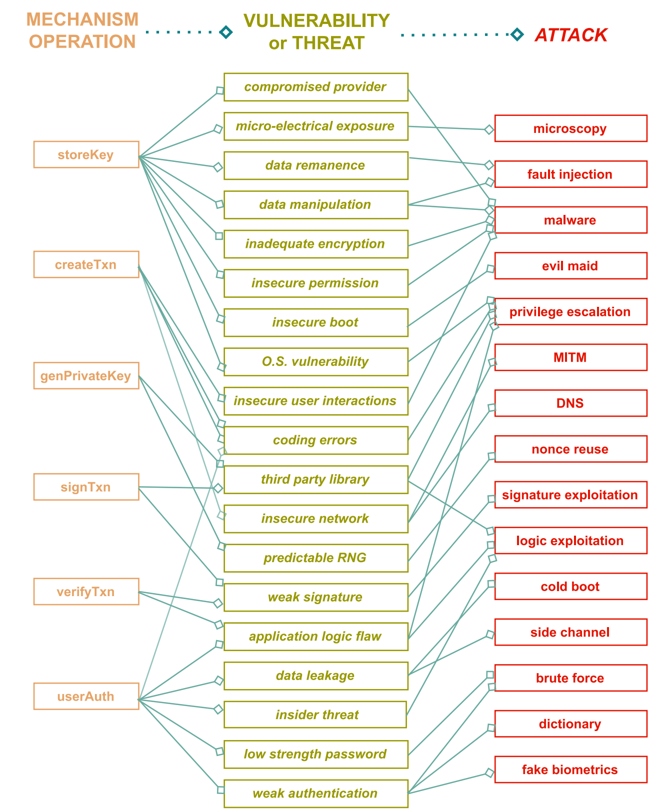

Fig. 4. Mapping of the wallet mechanism (Section 3) to threats/vulnerability occurrences (Section 6) and attack methods (Section 7).

7. Attack taxonomy

In this section, we present a comprehensive taxonomy of wallet attack vectors, systematically examining the methods, techniques, and targeted components involved. Building on our generalised wallet mechanisms and threat model taxonomy, we outline a broad spectrum of attacks, as illustrated in Fig. 5. These attacks are categorised according to the specific functions and components targeted within the wallet infrastructure (see Section 3) and the threats exploited (see Section 6.1). We further incorporate the infrastructure layer of our design taxonomy to capture the multi-layered nature of these threats, as summarised in Table 5.

7.1. Network

7.1.1. Connection hijack

These attacks aim to compromise the communication channel between wallets and other network participants using man-in-the-middle (MITM) attacks to intercept and modify the *txn* message generated by Algorithm 2. Various types of MITM attacks include Rogue AP [130], DNS spoofing [151,152], IP spoofing [146], and Border Gateway Protocol (BGP) hijacking [153], as shown in Table 5. Hardware wallets are vulnerable to these attacks if the online wallet client (see Section 5.1.2) is compromised. Ledger has previously reported susceptibility to MITM attacks.

The Rogue access point (AP) vector functions through unauthorised WiFi hotspots that can intercept transactions by exploiting the *txnInit* function. This allows an attacker to modify *state_trans_info* before blockchain forwarding, potentially redirecting funds to a different address than the recipient's address [130]. The Domain Name System (DNS) spoofing vector occurs when a DNS resolver, which

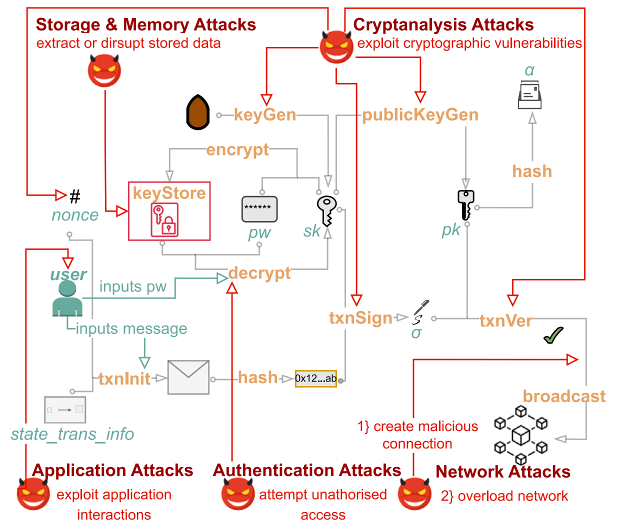

Fig. 5. Attack classification on wallet mechanism showing targeted operations and components (see Table 5).

translates human-readable domain names into IP addresses, is compromised [154]. This leads to fraudulent cryptocurrency service website redirection. One notable example is the 2017 EtherDelta DNS hijack, where attackers altered DNS records to redirect users to a phishing clone [155]. An attacker can also execute a Border Gateway Protocol (BGP) hijacking attack that maliciously advertises false BGP routes to divert traffic intended for legitimate blockchain nodes (see Algorithm 3) or wallet API endpoints [153]. The MyEtherWallet attacker employed the BGP hijacking vector [156]. Another connection hijack avenue, the Address Resolution Protocol (ARP) spoofing vector, is initiated when attackers broadcast fraudulent ARP messages across a local network. This links their MAC address with the IP address of a legitimate network host to redirect the user's transaction data generated in Algorithm 2 [130,153].

7.1.2. Service denial

This is executed using adversary-controlled devices to orchestrate distributed denial-of-service (DDoS) attacks which overwhelm the network infrastructure with an excessive volume of requests, causing a decline or cessation of wallet operations (see Section 3) [157]. These attacks often target the Internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP) handshake mechanism, and other network infrastructure [20]. One common medium of conducting a DDoS attack is through botnets, which involves an adversary using a network of computers [158].

The Internet Control Message Protocol (ICMP) flooding vector overloads a wallet with network requests (ICMP echo request packets) at a rate exceeding the processing capacity. This results in a decline or cessation of transaction management operations (see Section 3.3) [20]. An adversary may also disrupt the wallet network by exploiting the Transmission Control Protocol (TCP) handshake mechanism, which establishes a connection between the wallet application and its servers, through synchronise (SYN) attacks [20].

7.2. Application

7.2.1. Malware execution

This attack intrusively exploits system vulnerabilities to steal transaction data, the *sk* and password credentials, or to manipulate wallet operations as described in Section 3. Malware threatens the wallet mechanism by replacing the *recipient_address* via clipboard hijackers [39] or through input monitoring via keyloggers [138] and other

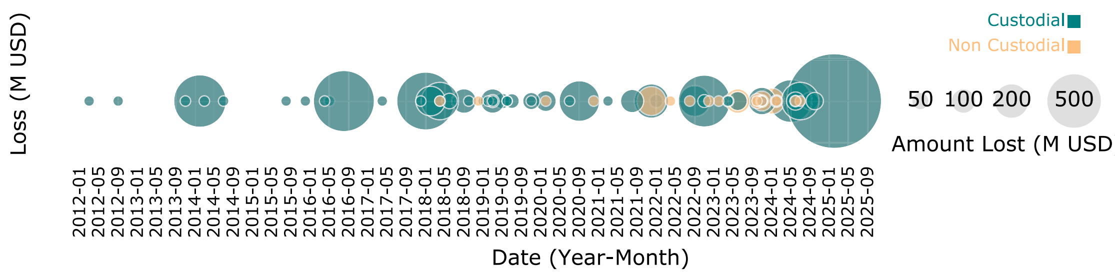

Fig. 6. Notable wallet incidents (in million USD) between 2012-01 and 2025-04, classified on the custody axis (Section 5.2.1). More detail is provided in Table 4.

spyware types [139,159]. Hardware wallets are also vulnerable to clipboard hijack attacks [102,160]; malware can be injected through interactions between the wallet and removable media such as USB drives [161].

Malware can also be engineered to monitor user actions and retrieve the user's password (*pw*) or private key (*sk*) [139,159]. Spyware includes keyloggers which can track every keystroke executed on an infected wallet device to steal confidential data [136,138]. The custodial wallet Cashaa [107] and non-custodial wallets BitKeep [162] and Bittensor [163] have previously been exploited by malware-based vectors. Malware can also be combined with other attack methods, such as social engineering or privilege escalation, to achieve hacks as noted in the ByBit case (see Section 9.1) (see Fig. 6).

7.2.2. Social engineering

These attacks aim to manipulate the user to divulge confidential data. Phishing attacks, for instance, aim to deceive wallet users into revealing *sk* or *pw* by mimicking legitimate services. Once successful, attackers can leverage additional vectors to gain unauthorised access [158].

Notably, malware delivered through phishing, such as Pink Drainer, Monkey Drainer, Venom Drainer, and Inferno, has been particularly effective against non-custodial wallets (see Table 4). Phishing attacks have also been effective against custodial wallets [164,165] and notable individuals [163]. Adversaries have also exploited third-party dependencies by targeting their personnel, thereby extending the reach of social engineering campaigns [133].

Telegram-embedded wallets heighten social-engineering exposure. Coordinated Telegram bots and rogue Mini App have drained millions from users [143,166]. Address-poisoning adds yet another twist: attackers inject look-alike addresses into a victim's history so that a routine copy-and-paste transaction quietly redirects funds [167].

7.2.3. Privilege escalation

These attacks aim to circumvent standard access controls to acquire elevated permissions. In Android root privilege attacks, the adversary can gain unauthorised root access to mobile wallets through vulnerabilities in the operating system (OS) [131]. Another OS-related attack, Android USB debugging [131], exploits operating system (OS) vulnerabilities in mobile devices by wireless debugging, using a computer connected to the same network. Following this, the adversary gains unrestricted access to manipulate the execution flow of the wallet and capture *sk*, *rdm_seed*, and other sensitive data [131].

7.2.4. Logic exploitation

Logic flow exploitation encompasses several wallet types and involves identifying and exploiting flaws in the programming logic of a wallet mechanism (Section 3) to gain unauthorised access or manipulate wallet functions [96]. Notable incidents include WazirX (2024), where investigators linked the drain to a malicious Safe module that slipped through the upgrade mechanism and rewired the wallet via `DELEGATECALL` [163]. In ByBit (2025), attackers pushed a forged implementation contract into the exchange's cold-wallet proxy, overwriting storage and seizing ownership by abusing Safe's upgrade path [15] (see Section 9.1). The classic Parity library bug (2017) involved an uninitialised contract that allowed the adversary to gain ownership and drain multi-sig wallets [95]. These cases map to two recurrent sub-patterns: (1) upgrade-path hijack, where the authorised proxy-upgrade or module-installation channel is abused to introduce attacker-controlled logic (ByBit, WazirX); and (2) constructor hijack, where the `init` function is left callable after deployment (Parity).

7.3. Authentication

7.3.1. Credential cracking

This category of attacks systematically attempts different credential values to bypass the authentication mechanism. Brute-force attacks involve an adversary systematically trying all possible character combinations to bypass the authentication function and decrypt *enc_sk*. If successful, the adversary can create malicious transactions using Algorithm 2 [135]. Dictionary attacks, on the other hand, leverage commonly used words to predict *rdm_seed* phrases for access. Unlike brute-force attacks that exhaust all possible combinations, dictionary attacks are computationally less demanding, and their success rate increases with the use of leaked password datasets [134,202].

7.3.2. Identity spoofing

For enhanced KEK security, wallets leverage supplementary user authentication methods, such as user biometrics and two-factor authentication (2FA) implementations.

The identity spoofing attack method bypasses these verification mechanisms (see Algorithm 1) by impersonating the user to decrypt *enc_sk* and authorise malicious transactions. In fake biometric attacks, an adversary employs synthetic or reconstructed biometric data to achieve this goal [203]. To circumvent SMS-based 2FA, an adversary can also use SIM swap attacks, which execute the transfer of the user's phone number to an adversary-controlled mobile device [204]. Mobile wallets, smart contract wallets and other infrastructures that integrate SMS-based 2FA or biometric verification can be vulnerable to these attacks (see Table 5).

7.4. Storage and memory

7.4.1. Physical tampering

These primarily involve physically altering a wallet's hardware to bypass security protections. In evil maid attacks, the attacker physically modifies the unencrypted storage of a device to capture credentials or manipulate the system [205]. In contrast, microscopy attacks use advanced techniques, such as electron microscopy, to examine the microelectronic components of a wallet. These attacks can extract critical data or identify vulnerabilities, often without altering the hardware itself [105].

Table 4
Wallet attack incidents in the industry. We retrieve 85 notable attack incidents involving both custodial and non-custodial wallets. Several attack methods remain unknown (–) or undetailed, we indicate undetailed incidents with * [168–199].

Name	Custody Design	Date	Loss ($)	Attack Category	Attack Name
ByBit [133]	Custodial	2025-02	1500M	Application	Logic exploitation
US Govt. [188]	Non-custodial	2024-10	50M	–	–
BigX [163]	Custodial	2024-09	52M	–	–
Indodax [179]	Custodial	2024-09	22M	–	–
WazirX [14]	Custodial	2024-07	235M	Application	Logic exploitation
Bittensor [163]	Non-custodial	2024-07	8M	Application	Malware
BTCTurk [163]	Custodial	2024-06	55M	–	–
Loopring [163]	Non-custodial	2024-06	5M	Authentication	Identity spoofing*
Lykke [107]	Custodial	2024-06	22M	–	–
DMM Bitcoin [163]	Custodial	2024-05	305M	–	–
Axie Co-Founder [188]	Non-custodial	2024-02	10M	–	–
Fixed Float [163]	Custodial	2024-02	26.1M	–	–
kirilm.eth [163]	Non-custodial	2024-02	5.1M	Application	Phishing
Ripple Co-Founder [186]	Non-custodial	2024-01	112.5M	–	–
HTX (Huobi) [164]	Custodial	2023-11	13.6M	–	*sk* compromise*
Pink Drainer [200]	Non-custodial	2023-11	12M	Application	Phishing, malware
Monkey Drainer [200]	Non-custodial	2023-11	16M	Application	Phishing, malware
Venom Drainer [200]	Non-custodial	2023-11	27M	Application	Phishing, malware
Infarno [13]	Non-custodial	2023-11	66M	Application	Phishing, malware
Poloniex [200]	Custodial	2023-11	126M	–	*sk* compromise*
Lastpass [200]	Non-custodial	2023-10	37M	Authentication	–
Fantom Fdn. [195]	Non-custodial	2023-10	7M	–	–
HTX (Huobi) [164]	Custodial	2023-09	8M	Application	Phishing
Fake Voucher [200]	Non-custodial	2023-09	4.5M	Application	Phishing
Remitano [200]	Custodial	2023-09	2.7M	Application	–
CoinEx [107]	Custodial	2023-09	55M	–	*sk* compromise*
Monero [184]	Non-custodial	2023-09	0.5M	–	–
AlphaPo [200]	Custodial	2023-07	60M	–	*sk* compromise*
Atomic Wallet [107]	Non-custodial	2023-06	100M	–	–
Bitrue [163]	Custodial	2023-04	23M	–	*sk* compromise*
GDAC [107]	Custodial	2023-04	13M	–	*sk* compromise*
MyAlgo [107]	Non-custodial	2023-02	9.2M	–	–
BitKeep [162]	Non-custodial	2022-12	8M	Application	Phishing, malware
FTX [178]	Custodial	2022-11	450M	Authentication	Sim swap attack
Deribit [177]	Custodial	2022-11	28M	Application	–
Wintermute [196]	Custodial	2022-09	160M	Authentication	Brute force
Slope [107]	Non-custodial	2022-08	8M	Storage and memory	–
MetaMask [162]	Non-custodial	2022-04	0.65M	Authentication	Phishing
Crypto.com [163]	Custodial	2022-01	30M	Authentication	–
Lympo [107]	Custodial	2022-01	18.7M	–	–
LCX [191]	Custodial	2022-01	8M	–	*sk* compromise*
Vulcan Forged [12]	Non-custodial	2021-12	140M	Application	*sk* compromise*
BitMart [165]	Custodial	2021-12	196M	Application	Phishing
Liquid [194]	Custodial	2021-08	90M	Application	*sk* compromise*
Roll [172]	Custodial	2021-03	5.7M	Application	*sk* compromise*
MetaMask [163]	Non-custodial	2020-12	8M	–	–
KuCoin [11]	Custodial	2020-09	275M	Application	*sk* compromise*
Cashaa [107]	Custodial	2020-07	3.1M	Application	Malware
Trinity Wallet [192]	Non-custodial	2020-02	2.3M	Application	–
Altsbit [199]	Custodial	2020-02	72.5M	Application	–
Upbit [201]	Custodial	2019-11	49M	Application	Phishing, malware
Bitpoint [183]	Custodial	2019-07	36.5M	–	–
Vindax [197]	Custodial	2019-11	0.5M	–	–
Bitrue [189]	Custodial	2019-06	4.5M	Authentication	–
Gatehub [198]	Custodial	2019-06	9.5M	–	–
Binance Exchange [187]	Custodial	2019-05	40M	Unknown	–
Bithumb [172]	Custodial	2019-03	13M	Other	Insider job
Coinbene [107]	Custodial	2019-03	99M	–	–
DragonEX [172]	Custodial	2019-03	1M	Application	–
Cryptopia [190]	Custodial	2019-02	16M	–	*sk* compromise*
LocalBitcoins [172]	Custodial	2019-01	0.02M	Application	Phishing
Electrum [175]	Non-custodial	2018-12	0.75M	Application	Phishing
Maplechange [171]	Custodial	2018-10	6M	–	–
Zaif [172]	Custodial	2018-09	100M	–	–
Coinrail [172]	Custodial	2018-06	40M	–	–
MyEtherWallet [156]	Non-custodial	2018-04	0.15M	Network	BGP hijacking
Gate.io [185]	Custodial	2018-04	234M	–	–
CoinSecure [172]	Custodial	2018-04	3.5M	Other	Insider job

(continued on next page)

Table 4 (*continued*).

Bitgrail [181]	Custodial	2018-02	146M	Other	Insider job
CoinCheck [180]	Custodial	2018-01	560M	–	–
BlackWallet [182]	Non-custodial	2018-01	0.4M	Network	DNS spoofing
EtherDelta [155]	Custodial	2017-12	1.4M	Network	DNS spoofing
Parity [95]	Non-custodial	2017-07	30M	Application	Logic exploitation
Yapizon [107]	Custodial	2017-04	5.3M	–	–
Bitfinex [172]	Custodial	2016-08	623M	Application	–
Gatecoin [172]	Custodial	2016-05	2.1M	–	–
Shapeshift [170]	Custodial	2016-04	0.23M	Other	Insider job
Bitstamp [174]	Custodial	2015-12	5M	Application	Phishing
BTER [172]	Custodial	2015-08	1.65M	Application	–
Mintpal [193]	Custodial	2014-07	2M	Other	Insider job
Poloniex [173]	Custodial	2014-03	0.05M	Application	–
Mt. Gox [10]	Custodial	2014-02	460M	–	–
Bitcash [176]	Custodial	2013-11	0.1M	Application	Phishing
Bitfloor [168]	Custodial	2012-09	0.25M	Application	*sk* compromise*
Bitcoinica [169]	Custodial	2012-03	0.09M	Application	*sk* compromise*
Summary:	**85 incidents**	**2012–2025**	**6.98B**		

7.4.2. Fault injection

These attacks manipulate the wallet's components by forcing an erroneous system state to bypass the security mechanisms [102]. For instance, fault injection attacks on hardware wallets often exploit vulnerabilities in volatile memory (such as SRAM) by manipulating environmental factors. Data remanence vulnerabilities in the Trezor wallet have been exploited to demonstrate these attacks [81,82]. Fault injection attacks on smart contracts have also been shown in the literature [103].

7.4.3. Other non-invasive techniques

Other non-invasive storage and memory attacks exist which are not based on fault injection methods. In cold boot attacks, the attacker executes a cold restart on the wallet device to exploit the data remanence properties of volatile memory, such as dynamic random-access memory (DRAM) and static random-access memory (SRAM), to retrieve sensitive data [136]. Similarly, PUF attacks exploit the unique characteristics of hardware defence implementations known as physically unclonable function (PUF). These implementations have challenge-response functionality that exhibits physical unclonability [104,206].

7.5. Cryptanalysis

7.5.1. Side-channel analysis

Non-invasive key extraction attacks on cryptographic functions, including timing and power side-channel analysis (SCA), are executed by exploiting side channels. These attacks exploit leakages in behaviours exhibited by cryptographic functions (see Section 3) through side channels to measure and extract values such as time and power [136,148]. Timing-based SCA measures the cryptographic function execution time. Successful implementation of a timing-based side-channel attack has been demonstrated on a Trezor One hardware wallet [149]. Power-based SCA analyses the cryptographic function's power trace, including the hash function. SCA on the hash function has been utilised to extract the *rdm_seed* [207].

7.5.2. Direct exploitation

These attacks directly target implementation errors within the cryptographic surface area. Weak signature (σ) attacks, for example, target weaknesses in the signing algorithm due to improper implementation, weak or outdated cryptographic algorithms or errors in encryption logic [137]. In addition, an adversary can exploit vulnerabilities in Algorithm 2 by reusing a nonce during transaction authorisation [147]. Such reuse can compromise the security of wallets by resulting in *sk* leakage [208].

8. Security measures

This section builds upon the framework outlined in Section 7 by presenting mitigation approaches against wallet attacks. We aim to examine defence mechanisms for each identified attack vector affecting wallets.

8.1. Network

Suspicious network activity can be detected through machine learning techniques, including anomaly detection models [209] and classification algorithms [138]. Additionally, dynamic network parameter adjustments [210] and other intrusion detection mechanisms [161,211] further contribute to identifying such anomalies.

To mitigate these attacks, wallets can adopt network security protocols that validate and authenticate IP addresses [229] and incorporate additional security layers within the wallet's network to prevent potential *txn* modification attempts by adversaries [224]. To limit or prevent distributed denial-of-service (DDoS) attacks, wallets must distinguish malicious and authentic network traffic using classifiers such as the decision tree algorithm [230] and reinforcement learning approaches to analyse patterns in network data [225]. Another mitigation approach involves analysing the network for unusual patterns, such as repeated request attempts from the same IP address [226].

8.2. Application

To mitigate the risk of message alteration by clipboard hijackers, wallets can employ features such as NFC and two-dimensional codes to prevent recipient address modification during transaction creation [39]. From a user perspective, human-readable addresses such as ENS [231] aid in detecting address tampering, though they have certain security vulnerabilities [232]. Wallets can also prevent system behaviour modifications by addressing specific attack vectors. Attack vectors that attempt these modifications by targeting vulnerabilities in the OS can be mitigated by employing code obfuscation [227] and runtime protection mechanisms [228]. Furthermore, by enforcing Control Flow Integrity (CFI) measures, wallets can ensure that control flow cannot be hijacked to deviate from intended control flow paths for malicious transactions [233].

8.3. Authentication

Wallets can incorporate features either as direct protection against specific attack methods or as general authentication bypass protection. By directly integrating improved functionalities to obstruct access to

Table 5
Three-level attack classification showing gap analysis, threat occurrences, adversary's target and mapping to possible security measures (Section 8). The "Gaps" summary shows that academic literature covers 24 of the 28 enumerated attack vectors (86%), whereas publicly reported incidents cover 9 vectors (32%). Notable incident percentages are calculated from a total of 85 reported industry incidents (see Table 4). Symbols: (●: include, ◐: part-inclusion (influenced by other factors), ○: not include) [212–223].

Category	Method	Vector	Threat																					Target																Goal			Infrastructure					Gaps		Possible Defence
																								Data							Mechanism						Other													
			Predictable RNG [72,73,147]	Inadequate authentication [134]	Inadequate encryption [64]	Application logic flaw [96,129,217]	Low-strength passwords [41,135]	Data leakage [78–80]	Data remanence [81,82]	Data manipulation [81,82]	Insecure Boot Environment [136]	Microelectronic component exposure [105]	Weak signature [137]	Inadequate signature verification [74,223]	Insecure permissions [70,71]	Library vulnerability [68,69]	OS vulnerabilities [131]	Coding errors [96]	Insec. Network [65–67]	Insec. User Interactions [83,84]	Comp. Provider [87]	Malicious insider [128]	Compromised insider [69]	Private key (sk)	Signature (σ)	Mnemonics (rdm_seed)	KEK or Password (pw)	Memory	State Trans. Info.	Nonce	KeyGen	UserAuth	KeyStore	TxnInit	TxnSign	TxnVer	Service Provider	Operating System	Wallet user	Transaction alteration	Credential compromise	Network disruption	Desktop wallet	Browser wallet	Mobile wallet	Smart wallet	Hardware Wallet	Academic papers No. (%)	Notable incidents No. (%)	
Network	Connection Hijack	Rogue AP [130]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●	○	○	○	○	○	○	○	●	○	○	○	○	●	○	●	●	○	●	●	○	○	●	●	●	●	●	1	0	[211,224]
		DNS spoofing [152,154]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●	○	○	○	○	○	○	○	●	○	○	○	○	●	○	●	●	○	●	●	○	○	●	●	●	●	●	2	3	[151,211,224]
		IP spoofing [146]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●	○	○	○	○	○	○	○	●	○	○	○	○	●	○	●	●	○	●	●	○	○	●	●	●	●	●	1	0	[211,214,224]
		BGP hijacking [153]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●	○	○	○	○	○	○	○	●	○	○	○	○	●	○	●	●	○	●	●	○	○	●	●	●	●	●	1	1	[153]
	Service Denial	ICMP flooding [20,216]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	◐	●	○	○	○	○	●	●	●	●	●	●	2	0	[214,225]
		TCP SYN flooding [20]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	◐	●	○	○	○	○	●	●	●	●	●	●	1	0	[225,226]
Application	Malware Execution	clipboard hijack [39,160,221]	○	○	●	○	○	○	○	●	○	○	○	○	●	○	●	○	○	●	○	○	○	○	○	○	○	○	●	○	○	○	○	●	○	○	○	●	●	●	○	○	●	●	●	●	●	3	8	[39,159]
		Spyware [139,212]	○	○	●	○	○	●	○	○	○	○	○	○	●	○	●	○	○	●	○	○	○	●	○	●	●	○	○	○	●	○	●	○	○	○	○	○	●	○	●	○	●	●	●	○	○	2		[159]
	Logic Exploitation	Constructor hijack [95]	○	○	○	●	○	○	○	○	○	○	○	○	●	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	●	○	0	1	[95]
		Upgrade-path hijack [15]	○	●	○	●	○	○	○	○	○	○	○	○	○	●	○	○	○	●	●	○	○	○	○	○	○	○	●	○	●	○	○	●	○	○	○	○	●	○	●	○	○	○	○	●	●	0	2	[15]
	Privilege Escalation	Android root privilege [131]	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●	○	○	○	○	○	○	●	○	●	○	○	○	○	●	○	●	○	○	○	○	●	○	○	●	○	○	○	●	○	○	1	0	[227]
		Android USB debugging [131]	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●	○	○	○	○	○	○	○	○	●	●	○	○	○	●	●	○	○	○	○	○	●	○	○	●	○	○	○	●	○	○	1	0	[39,228]
	Social Engineering	Phishing [31]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	●	●	●	○	●	●	○	○	○	○	●	○	○	○	○	○	○	●	○	●	○	●	●	●	●	●	1	15	[114,115,213]
		Address poisoning [167]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	●	○	○	○	○	●	○	○	○	○	●	○	○	○	●	●	●	●	●	0	1	[218]
Authentication	Credential Cracking	Brute-force [41,135,215]	○	●	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	○	○	○	○	●	○	○	○	○	○	○	○	○	●	○	●	●	●	○	●	3	0	[135,215]
		Dictionary [134,202]	○	●	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	●	○	○	○	○	○	○	○	○	●	○	●	●	●	○	●	2	0	[212]
	Identity Spoofing	Fake biometrics [203]	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	◐	○	○	○	○	●	○	○	○	○	○	○	○	○	●	○	○	○	●	●	○	1	0	[203]
		SIM Swap [204]	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	◐	○	○	○	○	●	○	○	○	○	○	○	○	○	●	○	○	○	●	●	○	0	1	[204]
Storage	Fault Injection	Fault injection attacks [102,103]	○	○	○	○	○	○	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	●	○	○	○	○	●	○	○	○	○	○	○	○	●	○	●	●	●	●	●	2	0	[144,222]
	Physical Tampering	Evil maid [136]	○	●	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	●	○	○	○	○	●	○	○	○	○	○	○	○	●	●	○	○	○	○	○	●	1	0	[213]
		Microscopy [105]	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	●	○	○	○	○	○	○	○	●	○	○	○	○	○	●	1	1	[44,220]
	Non-invasive Manip.	Cold boot attack [136]	○	○	○	●	○	○	○	○	●	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	●	○	○	○	○	○	○	○	●	○	●	●	●	○	●	1	0	[205]
		PUF attacks [104]	●	●	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	●	○	○	○	○	○	○	○	●	○	○	○	○	○	●	1	0	[148,207]
Cryptanalysis	Side-channel Analysis	Timing-based [149]	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	●	○	○	○	●	●	○	○	○	○	●	○	●	●	●	○	●	1	0	[102,219]
		Power on Crypt. Algo. [148]	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	●	○	○	○	●	●	○	○	○	○	●	○	●	●	●	○	●	1	0	[102,219]
		Power on Hash [207]	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	●	○	○	○	○	○	●	○	●	●	●	○	●	1	0	[102,219]
	Direct Exploitation	Weak signature [137]	●	○	○	○	○	○	○	○	○	○	●	●	○	●	○	●	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	●	○	●	●	●	○	●	1	0	[147]
		Nonce reuse [147]	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	○	○	○	●	○	○	○	○	○	●	○	●	●	●	○	●	1	0	[147]
	Summary	28 attack vectors																																													Attack vectors occurrence	24(86%)	9(32%)	

predictive text data, wallets can prevent dictionary attacks [134]. Additionally, to prevent brute-force attacks, only complex passwords should be allowed in the initialisation stage [202]. Biometric falsifying attacks can be prevented by incorporating liveness detection features in wallets [203].

To prevent single points of failure, wallets can enhance authentication levels (Section 5.5) through multi-factor authentication (MFA), multi-party computation (MPC) [115] and multi-signatory features such as BIP-11's M-of-N standard [114] (Section 5.4). To mitigate social engineering attacks, wallets can incorporate phishing-resistant MFA techniques such as FIDO2 [234]. This feature enables communication with the original wallet website to verify authenticity before allowing access to the wallet [235].

8.4. Storage and memory

An effective defence method against these attacks involves incorporating physically unclonable function (PUF) to generate cryptographic keys on demand, without storing *sk* on the wallet's chip. This method also prevents microscopy attacks, some other physical tampering attacks, and side-channel attacks (see Section 8.5) [44,207]. Physical tampering through the evil maid attack can be limited by implementing trusted boot mechanisms [236]. Possible mitigations against non-invasive manipulation, such as the cold boot attack, involve adopting features which algorithmically clear the wallet's memory following intrusion [237]. For example, Ledger has introduced a secure layer which detects chip intrusion and erases *sk* following extraction attempts [238].

8.5. Cryptanalysis

Exploiting cryptographic vulnerabilities can lead to *sk* extraction. Attacks that aim to exploit weak cryptographic signatures (σ) can be counteracted by employing stronger hashing algorithms [137], while deterministic *nonce* selection prevents nonce reuse attacks [147]. Non-invasive attacks on cryptographic functions, including timing and power SCA, are executed by exploiting side channels. Effective prevention methods include data leakage protection and disguising data access patterns as noise injection [102,207,239,240]. These affect the adversary's ability to interpret leaked information effectively [241].

9. Case studies

In this section, we present detailed case studies of notable wallet security breaches. We apply our wallet design taxonomy (Section 5), threat model (Section 6), and attack taxonomy (Section 7). Each case study systematically analyses the wallet's architecture, identifies exploited vulnerabilities, and explores the sequence of attack events. We conclude each study with recommended and implemented security measures.

9.1. Case study: ByBit custodial wallet hack

In February 2025, ByBit experienced a significant security breach that resulted in a loss of approximately $1.5 billion in Ethereum, marking the largest cryptocurrency theft to date [133]. This sophisticated attack aligns with the attack vectors outlined by our taxonomy. We provide a detailed analysis below using our frameworks for design classification, threat assessment, attack sequence analysis, and mitigation strategies.

9.1.1. Design

Using our design taxonomy in Section 5, we analyse the ByBit wallet design below:

- **Custody:** ByBit maintained full custody of user funds, with users relinquishing *sk* control to the exchange. This particular case pertains to the *sk*, which controlled the Ethereum assets of the exchange.
- **Infrastructure:** ByBit employed a multi-faceted infrastructure design, integrating hardware wallets with a smart contract-enabled proxy architecture. The primary proxy contract delegated logic execution to a separate implementation contract via `delegateCall`. It stored the implementation contract's address in storage slot 0 to facilitate future upgrades [242]. However, the design did not enforce strict access controls on this critical operation. This became a key factor exploited in the attack, as described in the threat analysis (see Section 9.1.2).
- **Distribution:** *sk* management was distributed securely with authorisation rights shared among multiple private key (*sk*) holders in the multi-sig scheme across different hardware devices. The multi-signature scheme prevented unilateral transactions, mandating consensus among multiple trusted individuals.
- **Authorisation:** Transactions were generated via Safe's web interface. Signers reviewed transaction details on the web user interface and hardware wallet screens. Only after confirmation on their Ledger hardware wallet devices were transactions broadcast to the blockchain.
- **Validation:** After obtaining the necessary approvals, transactions underwent validation to ensure compliance with ByBit's internal security policies. This included verifying adherence to address whitelisting protocols and transfer limits. The multi-sig smart contract enforced these policies by executing transactions only when the requisite number of valid signatures was present.

9.1.2. Threats and dependencies

ByBit's security architecture relied significantly on several interconnected elements, including the Safe user interface, which proved vulnerable to the adversaries' attempts. We outline the threats, which were exploited by the adversary inline with our threat model below:

- **Insecure Interaction:** Insecure interactions resulted in the system's exposure to threats. The adversary likely exploited these interactions to achieve infiltration of the Safe developer's machine [15].
- **Application Provider Compromise:** ByBit's operational security was heavily dependent on the integrity and security posture of third-party service providers, in this case, Safe's web interface.
- **Data Misrepresentation:** The adversary compromised the accuracy and reliability of transaction data presented to authorised signers through Safe's user interface. This highlighted a critical vulnerability in wallet user interfaces.
- **Application Logic Flaw:** The infrastructure design permitted unrestricted use of the `delegateCall` instruction, allowing malicious actors to overwrite critical storage slots. Specifically, the attackers exploited the ability to overwrite the logic pointer stored in storage slot 0, leading to unauthorised control of the proxy's logic [15]. This violated the principle of least privilege and directly facilitated the privilege escalation step of the attack.
- **Blind Signing:** ByBit's reliance on hardware wallet confirmation processes did not sufficiently address the blind signing risk. Signers assumed the hardware wallet displays were a trustworthy verification source and approved transactions without explicit visibility into critical transaction metadata. This included `delegateCall` operations and underlying implementation changes.

9.1.3. Adversary goal and capabilities

$\mathcal{A}$ aimed to gain unauthorised rights by masking adversary-created transactions as benign. The capabilities of $\mathcal{A}$ significantly evolved during the attack as extended knowledge was gained, starting from restricted external knowledge and progressing to insider-level knowledge and access:

- **Initial Phase:** A remotely exploited publicly accessible information to exploit Safe developer interactions and gain restricted internal access.
- **Intermediate Phase:** Having achieved insider-level knowledge and privileges following a successful repository compromise, A could inject malicious software into operational components of the wallet software.
- **Final Phase:** A could exploit application logic to deceive sk holders, achieving credential compromise. Subsequently, A gained full wallet control and authorisation rights.

9.1.4. Attack sequence

The ByBit incident represents a sophisticated combination of several coordinated attack vectors identified in our Application threats taxonomy:

- **Social Engineering:** A phishing attack method enabled the execution of subsequent attack vectors. Social engineering and malware were combined to compromise ByBit, as seen in past incidents (e.g., BitKeep [162], Upbit [201], and wallet drainers [200]). This gave the adversary direct access to Safe's front-end code repository, highlighting the importance of secure developer environments.
- **Malware Execution:** The compromised machine enabled the injection of malicious JavaScript into Safe's front-end code, targeting the transaction approval interface. The malware modified the transaction data displayed to sk holders. While legitimate transaction details were displayed in the Safe wallet user interface, the data sent to the hardware wallet was altered.
- **Privilege Escalation:** The approved transaction altered the smart contract's logic. The attackers exploited storage slot hijacking by crafting a transaction that used `delegateCall` to execute a spoofing contract. This contract's `transfer()` function wrote the attacker's malicious implementation address to storage slot 0 via the Ethereum Virtual Machine (EVM) `SSTORE` opcode, overwriting the proxy's logic pointer. With the proxy now delegating to the attacker's contract, all subsequent transactions executed attacker-controlled code in the proxy's context, granting full authorisation rights.

9.1.5. Security measures

Before the breach, ByBit used a layered security model: most funds were in a Safe contract, private keys on six Ledger devices, requiring 4-of –6 multi-sig. These measures were bypassed. After the incident, industry experts highlighted the following additional controls:

- **Independent Transaction Hash Verification:** The use of tools such as `safe-tx-hashes` to independently verify transaction hashes against on-chain data mitigates the risk of UI-level deception [243]. By enabling signers to cross-reference actual transaction payloads outside of potentially compromised interfaces, this approach detects malicious operations such as unauthorised `delegateCall` or logic pointer overwrites before execution.
- **Transaction Policy Enforcement via On-Chain Gatekeeping:** Preventative solutions such as Halborn's Seraph simulate signed transactions before execution and block operations that violate predefined organisational policies [133]. In the context of the ByBit attack, this approach could have flagged and halted the unauthorised upgrade triggered by the malicious `delegateCall`, enforcing a secondary layer of validation beyond signer intent.
- **Hardware Wallet Clear-Signing:** Require devices that support the on-device display of the complete destination, value, function selector, and raw calldata (clear-signing) before approval. This enables signers can independently verify every field and avoid hash-only blind signing, a weakness exploited in the ByBit breach [242].
- **Wallet Auditing:** Conducting regular audits focusing on storage layout consistency and `delegateCall` whitelisting and other wallet-related code is pertinent [132]

9.2. Case study: Slope non-custodial wallet hack

In August 2022, Slope Wallet experienced a severe security incident, resulting in the compromise of over 9200 user wallets on the Solana blockchain and a loss of approximately $4.1 million in SOL and USDC [87]. We provide a detailed analysis below using our frameworks for design classification, threat assessment, attack sequence analysis, and implemented security measures.

9.2.1. Design

Applying our design taxonomy, we analyse the Slope wallet design below:

- **Custody:** Slope utilised a non-custodial model where users retained complete control over the private key (sk). This case pertains to the management and leakage of the user's private key.
- **Infrastructure:** Slope used a mobile software wallet that relied on a self-hosted Sentry monitoring stack [244,245]. This setup collected application data for debugging but inadvertently logged sensitive information due to a faulty logging function.
- **Distribution:** Slope used a single-distribution model, with all cryptographic operations and storage conducted solely on the user's mobile device. No advanced key distribution methods, such as MPC or multi-signature schemes, were integrated.
- **Authorisation and Validation:** The standard Solana *Ed25519* signature flow was executed locally on the user device. Transaction broadcasting was performed via Slope's own RPC endpoints.

9.2.2. Threats and dependencies

Slope's security architecture relied heavily on interconnected dependencies, particularly its integrated application-monitoring stack, as detailed below:

- **Application-Monitoring Dependency:** Slope utilised an on-premise implementation of the Sentry SDK, designed to assist developers in debugging. A single improperly added `toString()` method circumvented built-in security filters, resulting in sensitive wallet private keys being unintentionally logged in plaintext [244].
- **Data Leakage:** Multiple defensive measures were used (collection filtering, Transport Layer Security (TLS) certificate pinning, database encryption at rest). However, collection filtering and database encryption were disabled, causing plaintext private keys to be stored in the database.
- **Third-Party Supply-Chain Threat:** Slope employed a self-hosted version of the third-party monitoring solution (Sentry), inheriting risks associated with configuration drift, patch management latency, and internal operational errors. This on-premise deployment introduced vulnerabilities typically mitigated by a SaaS-managed setup.
- **Insecure User Interaction:** Users continued to interact with wallets whose keys had potentially been exfiltrated. No built-in key-rotation prompt existed.

9.2.3. Adversary goal and capabilities

The adversary, A, aimed primarily for credential compromise, specifically targeting the user's private key (sk). The capabilities leveraged by A included:

- **Initial Phase:** A used knowledge of Slope's logging vulnerability (via reverse engineering or insider information) to target the timeframe and method to extract logged private keys.

- **Intermediate Phase:** $\mathcal{A}$ employed remote network access, either directly to the internal database or by intercepting TLS traffic prior to 18 July 2022. This remote capability allowed the extraction of plaintext private keys despite the security measures initially in place.
- **Final Phase:** $\mathcal{A}$ employed legitimate wallet signing authority using stolen keys and subsequently drained user funds directly via standard blockchain transactions without triggering conventional anomaly detection.

9.2.4. Attack sequence

In this incident, the adversary employed the logic exploitation vector to compromise credentials, as summarised below:

- **Logic Bug Introduction:** Slope utilised a helper function (`toString()`) to streamline debugging, unintentionally bypassing established security filters. This bug directly caused private keys to enter plaintext logging pipelines.
- **Data Pipeline Restart:** Slope utilised Kafka for data processing. After restarting, Kafka inadvertently flushed cached logs containing private keys in plaintext format directly into a PostgreSQL database.
- **Log Exfiltration:** $\mathcal{A}$ accessed the misconfigured Sentry instance and retrieved the plaintext seed phrases, fully compromising user private keys.
- **Wallet draining:** $\mathcal{A}$ utilised legitimate signing authority gained from compromised private keys and drained assets from 9229 wallet addresses within seven hours.

9.2.5. Security measures

Following the Slope Wallet breach, the development team initiated several immediate reactive security measures. The team promptly disabled the self-hosted Sentry server within 15 min of identifying the vulnerability and advised users to transfer their assets to new wallets [246]. Additionally, audits conducted by OtterSec and Slowmist confirmed that sensitive data, including private keys, had been inadvertently logged [247,248]. In response, Slope removed all sensitive logging functionalities and implemented a `beforeSend` whitelist to filter out confidential information [249].

To prevent such incidents in the future, it is crucial to ensure that application monitoring tools, such as Sentry, are meticulously configured to exclude sensitive data from logs. This involves implementing stringent data scrubbing protocols and avoiding the logging of private keys or seed phrases. Proper calibration of these safeguards is essential for preserving the confidentiality and integrity of user credential data.

10. Insights

We discuss insights on design, threats, attack methods, and security measures from academic papers, industry incidents, and case studies below:

10.1. Influence of design on threats

Despite a wide range of security setups, we observe that the majority of the design combinations of existing wallets surveyed have been threatened by multiple vulnerabilities, as shown in Table 2. This is due to similar implementations i.e., the use of replicated libraries and commonly integrated implementation proposals (e.g., EIP-4337). We also observe that some wallets have had numerous vulnerabilities discovered in industry and academia. Most notably, Ledger and Trezor have several data remanence, data manipulation and insecure cryptographic vulnerabilities. Furthermore, in mapping vulnerabilities to attacks, we observe that some vulnerabilities can lead to numerous attack vectors as shown in Fig. 4. These include inadequate authentication, insecure permissions, insecure user interactions, and particularly data leakage. The Slope Wallet incident exemplifies this, where an improperly configured debug logging mechanism led directly to private key leakage.

10.2. High occurrence of signature verification logic flaws

We observe that signature verification logic flaws account for the most vulnerability occurrences in various wallets surveyed, constituting 19%. Another interesting observation is the occurrence of this vulnerability in three diverse wallet security enhancement architectures, namely hardware, smart contract and MPC wallets [74–77].

10.3. Gap analysis on wallet threats

Conducting a gap analysis across industry and academic reports is difficult because many incidents do not disclose precise attack methods. We generally observe a high correlation between identified threats in industry and academia, except for insider and external threats. Specifically, in the following threats: malicious insider, compromised insider and compromised service provider threats. Although several custodial designs have been proposed by academia along with threat models, an investigation into the potential external threats and attacks in custodial setups would be highly beneficial for the industry. Notably, most industry attacks target exchanges and other custodial setups, as large funds are concentrated within a few wallet addresses. Additionally, research into these areas will also be pertinent due to the fact that wallet designs are gradually evolving into shared-custodial or other setups which require authentication from a centralised party (e.g., passkey, 2FA).

To address the gaps identified in Table 3, we propose the following measures:

- **Responsible Disclosure Policies:** Create a standardised incident template for responsible disclosure of wallet-related incidents. This could employ a uniform reporting format for exchanges and custodians to use when disclosing incidents, enabling both industry and academic audiences to analyse them consistently. A notable example in industry is Immunefi's vulnerability disclosure platform [250].
- **Public-Private Collaborations:** Formalise partnerships between exchanges, blockchain security firms, and academic institutions to analyse incident data. Successful models exist, including as IC3 and Chainlink partnership [251] and the Stanford Centre for Blockchain Research's industry partnerships [252].
- **Open-source Incident Registry:** Develop an open repository where vetted blockchain incident post-mortems can be deposited by operators and accessed by researchers, policymakers, and other exchanges. An existing example is the SlowMist Hacked incident archive [253].

10.4. Difference in academia and notable industry incidents

Identifying attack vectors within the industry remains challenging, as sources often lack specificity. Notable attack vectors are significantly less clear (46% unknown) and show a lower spread compared to attacks described in the literature (see Table 5). This might be attributed to a lack of detailed post-mortem analysis in several incidents and an adversary's tendency to prioritise cost-effective methods. Academia, on the other hand, shows a high percentage (93%) and spreads across various attack methods. Our case study on the ByBit incident also exemplifies the complexity of real-world incidents compared to academic models. While academic literature often isolates attack vectors, the ByBit incident involved a multi-stage, multi-vector attack with a chain of sub-goals linked to the main goal of sk compromise.

Table 6
Defence methods categorised by type showing classification frequency (#) and percentage (%). Precautionary methods proactively prevent attacks; remedial methods provide attack detection, response, or data recovery [254,255].

Classification		[224]	[151]	[214]	[225]	[226]	[39]	[159]	[227]	[255]	[213]	[212]	[203]	[205]	[144]	[44]	[219]	[147]	[207]	[102]	[115]	[114]	[148]	[254]	[204]	[222]	[211]	[228]	[218]	[220]	# (%)
		Possible defence methods																													
Precautionary	Prevention	○	○	○	○	○	●	○	○	●	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	3(10%)
	Protection	●	●	●	○	○	●	●	●	○	●	●	●	○	○	●	●	○	●	●	○	○	●	●	○	○	○	●	○	●	17(58%)
	Limitation	○	○	○	●	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	●	●	○	○	●	○	○	○	●	○	6(21%)
Remedial	Detection	○	○	○	●	○	○	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	○	●	○	●	●	○	○	○	5(17%)
	Response	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	1(3%)
	Recovery	○	○	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	1(3%)
Summary		Precautionary: 26(89%)									Remedial: 7(24%)													Total unique methods							29(100%)

10.5. High-risk third-party dependencies

The ByBit attack highlights a critical systemic risk in modern wallet architectures: third-party dependencies can nullify even highly secure solutions. Despite ByBit's use of hardware wallets, multi-sig authorisation, and transaction policies, its reliance on Safe's third-party UI created a single point of failure. Similarly, Slope Wallet's reliance on a self-hosted instance of a third-party monitoring solution (Sentry) introduced vulnerabilities due to misconfiguration and operational errors. This further underscores how third-party integrations significantly impact wallet security. This demonstrates that wallet security inherits the weakest link in dependency chains. To mitigate these risks, wallets must adopt resilient architectures and proactively manage third-party risks through multi-layered audits and adversarial scenario modelling.

10.6. Comparison of custodial and non-custodial attacks

Our incident analysis reveals that custodial wallets and non-custodial accounts for 70% and 30% of attacks, respectively. Additionally, unknown methods are significantly higher in custodial wallets (50%) than in non-custodial wallets (36%). Incidents show a high degree of similarity between custodial and non-custodial attacks. For instance, in comparison to other attacks, phishing attacks account for a relatively high percentage of both custodial (10%) and non-custodial (36%) wallets, especially factoring in the number of unknown attacks.

10.7. High malware and phishing attack occurrence

We also find that application attacks account for a significant percentage of incident occurrences (43%), with 34% in custodial wallets and 48% in non-custodial wallets. Our data also indicates that malware and phishing attacks are the most common attack vectors, accounting for 10% and 18% of total incidents, respectively. We also find that phishing-malware attacks constitute 48% of total non-custodial wallet attacks.

10.8. Limitations of security measures

The majority of defence implementations in academia are particularly tailored to specific advanced attacks such as PUF for microscopic attacks, correlation elimination sounds for non-invasive side channels, and PUF attacks. Despite this, academia does not account for sophisticated attacks, which may leverage multiple attack vectors. Furthermore, distributed architectures prevalent in the industry are insufficient if dependencies remain centralised. The ByBit breach demonstrates that security measures must extend to third-party components, requiring redundant safeguards such as on-chain transaction simulation to detect UI spoofing or logic hijacking. In addition, the Slope Wallet incident demonstrates how inadequate configuration of application monitoring tools can undermine otherwise secure implementations, highlighting the need for strict data scrubbing and monitoring configurations.

10.9. Comparison of precautionary and remedial defence methods

Our study presents defence methods applicable to various attack vectors, with the majority offering either precautionary or remedial strategies, as illustrated in Table 6. Notably, precautionary defences significantly outnumber remedial approaches, comprising roughly 89% of all methods observed. Within the precautionary category, protection-focused implementations are the most prevalent, accounting for 58%. Among remedial defences, detection methods are the most common at 17%, while response and recovery measures each represent a mere 3%. This disparity highlights a critical gap in reactive mitigation techniques, indicating a potential area for further development in response and recovery-focused defences.

11. Discussion

11.1. Limitations

One significant limitation of our study is the quality and completeness of the data available on wallet attacks. As highlighted, many recorded incidents from custodial and non-custodial wallet providers contain a high degree of uncertainty regarding attack vectors (see Table 4). This ambiguity restricts our capability to perform detailed quantitative analyses of wallet attacks, thereby limiting the precision of our analysis.

11.2. Future work

To address these limitations, we propose the following research directions to improve wallet security:

11.2.1. Enhanced transaction validation measures

Our study highlights the uncertainty in recorded attack vectors, underscoring the need for enhanced transaction validation approaches. Advanced validation methods, such as independent transaction hash verification and proactive policy enforcement through on-chain gatekeeping, should be explored to improve transaction data clarity and reliability. Furthermore, integrating hardware wallets capable of clear-signing raw transaction parameters will significantly mitigate risks associated with deceptive UI interactions and unauthorised operational logic.

11.2.2. Addressing signature verification logic flaws

Given the prevalence of signature verification logic flaws across wallet architectures, targeted research is crucial for developing secure and robust signature verification frameworks. Future work should prioritise the formal verification of signature verification algorithms, exploring cryptographic approaches specifically designed to mitigate known logic vulnerabilities. This will directly enhance the integrity and trustworthiness of wallet systems.

11.2.3. Development of reactive defence mechanisms

Our study identified a substantial gap in reactive security measures, with an evident imbalance favouring preventive strategies. Future research should emphasise the development of advanced reactive mitigation strategies, including real-time anomaly detection, responsive incident management protocols, and automated recovery frameworks tailored explicitly for wallet incidents. Enhancing reactive defence capabilities will substantially improve resilience and responsiveness to evolving threat vectors. By addressing these targeted research areas informed by our identified limitations, the community can significantly advance wallet security practices. This will lead to improved theoretical understanding and enhanced practical outcomes.

12. Conclusion

This paper systematically analyses the design, threats, attack vectors, and defensive strategies associated with cryptocurrency wallets. We introduce a comprehensive multi-dimensional taxonomy of wallet architectures, providing a structured and detailed framework to effectively understand and navigate the complex security landscape across various wallet types. By systematising diverse attack vectors, our framework offers clear insights into vulnerabilities and protective measures relevant to each wallet category.

Our analysis extends to examining 85 significant security incidents, accounting for financial losses exceeding $6.98 billion. Through this systematic review, we propose targeted mitigation strategies corresponding to identified attack vectors and informed by our design taxonomy and security framework. Furthermore, our mapping of wallet mechanisms to specific design choices, threat profiles, attack methodologies, and existing defensive implementations underscores the critical interplay between different security dimensions and elucidates best practices.

We conduct a comparative analysis of incidents documented in industry contexts and vulnerabilities identified in academic research, revealing key gaps and convergence points between practical security threats and theoretical understandings. To further illustrate the practical applicability of our taxonomy and framework, we conduct detailed case studies, demonstrating its effectiveness in analysing and mitigating real-world wallet vulnerabilities and attacks.

By presenting an integrated perspective combining theoretical insights with empirical findings, our work lays the foundation for future research and practical advances, significantly enhancing the security and reliability of cryptocurrency wallets.

Acronyms

2FA Two-factor authentication

AEAD Authenticated encryption with associated data

AES Advanced Encryption Standard

AP Access point

API Application programming interface

ARP Address Resolution Protocol

AUM Assets under management

BGP Border Gateway Protocol

BIP Bitcoin Improvement Proposal

CFI Control Flow Integrity

CVE Common Vulnerabilities and Exposures

DDoS Distributed denial-of-service

DeFi Decentralised finance

DeRec Decentralised Recovery

DNS Domain Name System

DRAM Dynamic random-access memory

ECDSA Elliptic Curve Digital Signature Algorithm

EdDSA Edwards-curve Digital Signature Algorithm

EIP Ethereum Improvement Proposal

ENS Ethereum Name Service

ERC Ethereum Request for Comments

EVM Ethereum Virtual Machine

GCM Galois/Counter Mode

HD Hierarchical deterministic

HMAC Hash-based message authentication code

HSM Hardware security module

ICMP Internet Control Message Protocol

IP Internet Protocol

KDF Key derivation function

KEK Key encryption key

MFA Multi-factor authentication

MFKDF Multi-factor key derivation function

MITM Man-in-the-middle

MPC Multi-party computation

NFC Near Field Communication

OS Operating system

PBKDF Password-based key derivation function

PUF Physically unclonable function

RNG Random number generator

RPC Remote procedure call

SCA Side-channel analysis

SE Secure element

SHA Secure Hash Algorithm

SLIP SatoshiLabs Improvement Proposal

SRAM Static random-access memory

SYN Synchronise

TCP Transmission Control Protocol

TLS Transport Layer Security

CRediT authorship contribution statement

Yimika Erinle: Writing – review & editing, Writing – original draft, Visualization, Methodology, Data curation, Conceptualization. **Yathin Kethepalli:** Writing – original draft, Methodology, Data curation. **Yebo Feng:** Writing – review & editing, Supervision, Investigation. **Jiahua Xu:** Writing – review & editing, Visualization, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Yebo Feng reports administrative support was provided by Ripple. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank Liyi Zhou and Zhipeng Wang for their in-depth review and constructive discussions during the study. This research is based upon work partially supported by University Blockchain Research Initiative (UBRI) [256]. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Ripple.

Data availability

The code and data supporting the findings of this study is openly available at: https://github.com/xujiahuayz/crypto-wallets.

References

[1] N. Satoshi, Bitcoin: A peer-to-peer electronic cash system, 2008.
[2] S. Arora, Y. Li, Y. Feng, J. Xu, SecPLF: Secure protocols for loanable funds against oracle manipulation attacks, in: Proceedings of the 19th ACM Asia Conference on Computer and Communications Security, 2024, pp. 1394–1405.
[3] J. Xu, K. Paruch, S. Cousaert, Y. Feng, Sok: Decentralized exchanges (dex) with automated market maker (amm) protocols, ACM Comput. Surv. 55 (11) (2023) 1–50.
[4] Y. Luo, Y. Feng, J. Xu, P. Tasca, Piercing the veil of tvl: defi reappraised, in: Financial Cryptography and Data Security (FC 2025), Miyakojima, Japan, 2025, https://fc25.ifca.ai/preproceedings/94.pdf.
[5] S. Cousaert, N. Vadgama, J. Xu, Token-based insurance solutions on blockchain, in: Blockchains and the Token Economy: Theory and Practice, Springer, 2022, pp. 237–260.
[6] S. Cousaert, J. Xu, T. Matsui, Sok: Yield aggregators in defi, in: 2022 IEEE International Conference on Blockchain and Cryptocurrency, ICBC, IEEE, 2022, pp. 1–14.
[7] M. Mita, K. Ito, S. Ohsawa, H. Tanaka, What is stablecoin?: A survey on price stabilization mechanisms for decentralized payment systems, in: 2019 8th International Congress on Advanced Applied Informatics, IIAI-AAI, IEEE, 2019, pp. 60–66.
[8] Y. Luo, Y. Feng, J. Xu, P. Tasca, Y. Liu, LLM-powered multi-agent system for automated crypto portfolio management, 2025, arXiv preprint arXiv:2501.00826.
[9] M.M.A. Khan, H.M.A. Sarwar, M. Awais, Gas consumption analysis of ethereum blockchain transactions, Concurr. Comput. Pr. Exp. 34 (4) (2022) e6679, http://dx.doi.org/10.1002/CPE.6679.
[10] Jimmy Song, Mt. Gox hack: Technical explanation, 2018, URL: https://jimmysong.medium.com/mt-gox-hack-technical-explanation-37ea5549f715.
[11] W.Z. Ada Hui, Over $280M drained in KuCoin crypto exchange hack, 2020, URL: https://www.coindesk.com/markets/2020/09/26/over-280m-drained-in-kucoin-crypto-exchange-hack/.
[12] CoinDesk, Vulcan forged play-to-earn gaming platform refunds users after $140M hack, 2021, URL: https://www.coindesk.com/business/2021/12/14/gaming-platform-vulcan-forged-refunds-users-after-140m-hack/.
[13] Infosecurity Magazine, Inferno drainer returns, stealing millions from crypto wallets, 2025, URL: https://www.infosecurity-magazine.com/news/inferno-drainer-returns-stealing/.
[14] Explained: the WazirX hack, 2024, URL: https://www.halborn.com/blog/post/explained-the-wazirx-hack-july-2024.
[15] Certik, Bybit incident technical analysis , 2025, URL: https://www.certik.com/resources/blog/bybit-incident-technical-analysis.
[16] N.T. Courtois, R. Mercer, Stealth address and key management techniques in blockchain systems, in: ICISSP 2017 - Proceedings of the 3rd International Conference on Information Systems Security and Privacy 2017-January, SciTePress, 2017, pp. 559–566, http://dx.doi.org/10.5220/0006270005590566.
[17] E.V. Mangipudi, U. Desai, M. Minaei, M. Mondal, A. Kate, Uncovering impact of mental models towards adoption of multi-device crypto-wallets, in: CCS 2023 - Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, Association for Computing Machinery, Inc, 2023, pp. 3153–3167, http://dx.doi.org/10.1145/3576915.3623218.
[18] S. He, Q. Wu, X. Luo, Z. Liang, D. Li, H. Feng, H. Zheng, Y. Li, A social-network-based cryptocurrency wallet-management scheme, IEEE Access 6 (2018) 7654–7663, http://dx.doi.org/10.1109/ACCESS.2018.2799385.
[19] M. Di Angelo, G. Slazer, G. Salzer, Characteristics of wallet contracts on ethereum, in: 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services, IEEE. Institute of Electrical and Electronics Engineers Inc, 2020, pp. 232–239, http://dx.doi.org/10.1109/ICBC48266.2020.9169467.
[20] R. Chaganti, R.V. Boppana, V. Ravi, K. Munir, M. Almutairi, F. Rustam, E. Lee, I. Ashraf, A comprehensive review of denial of service attacks in blockchain ecosystem and open challenges, IEEE Access 10 (2022) 96538–96555, http://dx.doi.org/10.1109/ACCESS.2022.3205019.
[21] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J.A. Kroll, E.W. Felten, Sok: Research perspectives and challenges for bitcoin and cryptocurrencies, in: IEEE Symposium on Security and Privacy, 2015, pp. 104–121.
[22] S. Eskandari, D. Barrera, E. Stobert, J. Clark, D. Barrera, E. Stobert, A first look at the usability of bitcoin key management, 2015, http://dx.doi.org/10.14722/usec.2015.23015, arXiv preprint arXiv:1802.04351.
[23] K. Karantias, Sok: A taxonomy of cryptocurrency wallets, in: Cryptology EPrint Archive, 2020.
[24] I. Homoliak, M. Peresini, SoK: Cryptocurrency wallets - A security review and classification based on authentication factors, in: 2024 IEEE International Conference on Blockchain and Cryptocurrency, ICBC 2024, Institute of Electrical and Electronics Engineers Inc, 2024, http://dx.doi.org/10.1109/ICBC59979.2024.10634439.
[25] X. Li, P. Jiang, T. Chen, X. Luo, Q. Wen, A survey on the security of blockchain systems, Future Gener. Comput. Syst. 107 (2020) 841–853, http://dx.doi.org/10.1016/J.FUTURE.2017.08.020.
[26] H. Guo, X. Yu, A survey on blockchain technology and its security, Blockchain Res. Appl. 3 (2) (2022) 100067, http://dx.doi.org/10.1016/J.BCRA.2022.100067.
[27] H. Chen, M. Pendleton, L. Njilla, S. Xu, A survey on ethereum systems security: Vulnerabilities, attacks, and defenses, ACM Comput. Surv. 53 (3) (2020) http://dx.doi.org/10.1145/3391195.
[28] L. Zhou, X. Xiong, J. Ernstberger, S. Chaliasos, Z. Wang, Y. Wang, K. Qin, R. Wattenhofer, D. Song, A. Gervais, Sok: Decentralized finance (defi) attacks, in: 2023 IEEE Symposium on Security and Privacy, SP, IEEE, 2023, pp. 2444–2461.
[29] W.M. Shbair, E. Gavrilov, R. State, HSM-based key management solution for ethereum blockchain, in: IEEE International Conference on Blockchain and Cryptocurrency, ICBC 202, Institute of Electrical and Electronics Engineers Inc, 2021, http://dx.doi.org/10.1109/ICBC51069.2021.9461136.
[30] J.S. Götte, B. Scheuermann, G. De, Tech report: Inerial HSMs thwart advanced physical attacks, Cryptol. EPrint Arch. (2021) http://dx.doi.org/10.13154/tches.v2020.i4.309-336.
[31] A.A. Andryukhin, Phishing attacks and preventions in blockchain based projects, in: 2019 International Conference on Engineering Technologies and Computer Science, EnT, IEEE, 2019, pp. 15–19.
[32] T. Bui, S.P. Rao, M. Antikainen, T. Aura, Pitfalls of open architecture: How friends can exploit your cryptocurrency wallet, in: Proceedings of the 12th European Workshop on Systems Security, 2019, pp. 1–6.
[33] P. Das, S. Faust, J. Loss, A formal treatment of deterministic wallets, Proc. ACM Conf. Comput. Commun. Secur. (2019) 651–668, http://dx.doi.org/10.1145/3319535.3354236/SUPPL{_}FILE/P651-DAS.WEBM.
[34] I. Eyal, On cryptocurrency wallet design, OpenAccess Ser. Inform. 97 (2022) http://dx.doi.org/10.4230/OASICS.TOKENOMICS.2021.4/-/STATS.
[35] S. Houy, P. Schmid, A. Bartel, Security aspects of cryptocurrency wallets - A systematic literature review, ACM Comput. Surv. (2023) http://dx.doi.org/10.1145/3596906.
[36] I. Homoliak, D. Breitenbacher, O. Hujnak, P. Hartel, A. Binder, P. Szalachowski, SmartOTPs: An air-gapped 2-factor authentication for smart-contract wallets, in: AFT 2020 - Proceedings of the 2nd ACM Conference on Advances in Financial Technologies, Association for Computing Machinery, Inc, 2020, pp. 145–162, http://dx.doi.org/10.1145/3419614.3423257.
[37] S. Suratkar, M. Shirole, S. Bhirud, Cryptocurrency wallet: A review, in: 4th International Conference on Computer, Communication and Signal Processing, IEEE, 2020, pp. 1–7.
[38] E. Zaghloul, T. Li, M.W. Mutka, J. Ren, Bitcoin and blockchain: Security and privacy, IEEE Internet Things J. 7 (10) (2020) 10288–10313.

[39] C. Li, D. He, S. Li, S. Zhu, S. Chan, Y. Cheng, Android-based cryptocurrency wallets: Attacks and countermeasures, in: 2020 IEEE International Conference on Blockchain (Blockchain), IEEE, 2020, pp. 9–16, http://dx.doi.org/10.1109/Blockchain50366.2020.00010, IEEE.
[40] W. Dai, J. Deng, Q. Wang, C. Cui, D. Zou, H. Jin, SBLWT: A secure blockchain lightweight wallet based on trustzone, IEEE Access 6 (2018) 40638–40648, http://dx.doi.org/10.1109/ACCESS.2018.2856864.
[41] T. Volety, S. Saini, T. McGhin, C.Z. Liu, K.-K.R. Choo, Cracking Bitcoin wallets: I want what you have in the wallets, Future Gener. Comput. Syst. 91 (2019) 136–143.
[42] A.G. Khan, A.H. Zahid, M. Hussain, U. Riaz, Security of cryptocurrency using hardware wallet and QR code, in: International Conference on Innovative Computing, 2019, pp. 1–10, http://dx.doi.org/10.1109/ICIC48496.2019.8966739.
[43] H. Rezaeighaleh, Improving Security of Crypto Wallets in Blockchain Technologies (Electronic Theses and Dissertations), 2020, 2020-. 403. 2020–403 URL: https://stars.library.ucf.edu/etd2020/403.
[44] P. Urien, Innovative countermeasures to defeat cyber attacks against blockchain wallets, in: 2021 5th Cyber Security in Networking Conference, CSNet 2021, Institute of Electrical and Electronics Engineers Inc, 2021, pp. 49–54, http://dx.doi.org/10.1109/CSNET52717.2021.9614649.
[45] H. Rezaeighaleh, C.C. Zou, Multilayered defense-in-depth architecture for cryptocurrency wallet, in: 2020 IEEE 6th International Conference on Computer and Communications, ICCC 2020, Institute of Electrical and Electronics Engineers Inc, 2020, pp. 2212–2217, http://dx.doi.org/10.1109/ICCC51575.2020.9345013.
[46] H. Rezaeighaleh, C.C. Zou, New secure approach to backup cryptocurrency wallets, in: 2019 IEEE Global Communications Conference, GLOBECOM, IEEE, 2019, pp. 1–6.
[47] P. Chatzigiannis, K.C. Wang, S. Arora, M. Minaei, A composability analysis framework for Web3 wallet recovery mechanisms, in: 2025 IEEE Symposium on Security and Privacy, SP, IEEE, 2025, pp. 1531–1546.
[48] Standards for Efficient Cryptography Group, Sec 2: Recommended elliptic curve domain parameters, 2010, Version 2.0, §2.7 (secp256k1).
[49] J. Josefsson, I. Liusvaara, Edwards-curve digital signature algorithm (EdDSA), 2017, RFC 8032.
[50] e.a. Pieter Wuille, BIP 32: Hierarchical deterministic wallets, 2012, https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki.
[51] SatoshiLabs, SLIP 10: Universal private key derivation, 2019, https://github.com/satoshilabs/slips/blob/master/slip-0010.md.
[52] M. Palatinus, P. Rusnak, A. Voisine, S. Bowe, BIP-0039: Mnemonic code for generating deterministic keys, 2013, URL: https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki#Generating_the_mnemonic. (Accessed 01 July 2025).
[53] SatoshiLabs, SLIP 39: Shamir backup for BIP-39 mnemonics, 2019, https://github.com/satoshilabs/slips/blob/master/slip-0039.md.
[54] e.a. Slush, BIP 44: Multi-account hierarchy for deterministic wallets, 2014, https://github.com/bitcoin/bips/blob/master/bip-0044.mediawiki.
[55] T. Voegtlin, BIP 49: Derivation scheme for P2WPKH-in-P2SH accounts, 2016, https://github.com/bitcoin/bips/blob/master/bip-0049.mediawiki.
[56] P. Swamy, BIP 84: Derivation scheme for native SegWit accounts, 2017, https://github.com/bitcoin/bips/blob/master/bip-0084.mediawiki.
[57] B. Kaliski, J. Jonsson, PKCS #5: Password-based cryptography specification version 2.1, 2017, RFC 8018.
[58] C. Percival, S. Josefsson, The scrypt password-based key derivation function, 2016, RFC 7914.
[59] D. Steele, F. Saarinen, L. Birkholz, A. Cohn-Gordon, The memory-hard Argon2 password hash and proof-of-work function, 2021, RFC 9106.
[60] NIST, Advanced Encryption Standard (AES), FIPS Publication 197, 2001.
[61] Y. Nir, A. Langley, Chacha20 and poly1305 for IETF protocols, 2018, RFC 8439.
[62] N.V. Database, CVE-2023-37192 detail, 2023, URL: https://nvd.nist.gov/vuln/detail/CVE-2023-37192.
[63] Open Zeppelin, Backdooring gnosis safe multisig wallets, 2020, URL: https://blog.openzeppelin.com/backdooring-gnosis-safe-multisig-wallets. https://blog.openzeppelin.com/backdooring-gnosis-safe-multisig-wallets.
[64] National Vulnerability Database, CVE-2019-15947 detail, 2019, URL: https://nvd.nist.gov/vuln/detail/CVE-2019-15947.
[65] National Vulnerability Database, CVE-2023-33297 detail, 2023, URL: https://nvd.nist.gov/vuln/detail/CVE-2023-33297.
[66] N.V. Database, CVE-2020-14198 detail, 2020, URL: https://nvd.nist.gov/vuln/detail/CVE-2020-14198.
[67] National Vulnerability Database, CVE-2018-17144 detail, 2018, URL: https://nvd.nist.gov/vuln/detail/CVE-2018-17144.
[68] Kaspersky, Vulnerability in hot cryptowallets from 2011 2015, 2023, URL: https://www.kaspersky.co.uk/blog/vulnerability-in-hot-cryptowallets-from-2011-2015/26984/.
[69] Ledger, Security incident report, 2023.
[70] National Vulnerability Database, CVE-2022-32969 detail, 2022, URL: https://www.cvedetails.com/cve/CVE-2022-32969/.
[71] Halborn, Demonic vulnerability, 2022, URL: https://www.halborn.com/disclosures/demonic-vulnerability.
[72] National Vulnerability Database, CVE-2023-31290 detail, 2023, URL: https://nvd.nist.gov/vuln/detail/CVE-2023-31290.
[73] N.V. Database, CVE-2024-23660 detail, 2024, URL: https://nvd.nist.gov/vuln/detail/CVE-2024-23660.
[74] National Vulnerability Database, CVE-2020-14199 detail, 2020, URL: https://nvd.nist.gov/vuln/detail/CVE-2020-14199.
[75] Fireblocks, Lindell17 abort vulnerability technical report, 2023, URL: https://www.fireblocks.com/blog/lindell17-abort-vulnerability-technical-report/.
[76] ChainLight, Account abstraction security guide | by ChainLight | ChainLight blog & research | medium, 2023, URL: https://medium.com/chainlight/patch-thursday-account-abstraction-security-guide-c348cc5e36ee.
[77] Braavos, Uncovering the critical argent-X wallet vulnerability, 2022, URL: https://braavos.app/zero-click-argent-x-wallet-contract-vulnerability-explained/.
[78] National Vulnerability Database, CVE-2019-14353 detail, 2019, URL: https://nvd.nist.gov/vuln/detail/CVE-2019-14353.
[79] N.V. Database, CVE-2019-14354 detail, 2019, URL: https://nvd.nist.gov/vuln/detail/CVE-2019-14354.
[80] Kraken, Kraken identifies critical flaw in trezor hardware wallets, 2020, URL: https://blog.kraken.com/product/security/kraken-identifies-critical-flaw-in-trezor-hardware-wallets.
[81] Trezor, Fixing physical memory access issue in trezor, 2017, URL: https://blog.trezor.io/fixing-physical-memory-access-issue-in-trezor-2b9b46bb4522.
[82] Medium, Frozen trezor data remanence attacks, 2017, URL: https://medium.com/@Zero404Cool/frozen-trezor-data-remanence-attacks-de4d70c9ee8c.
[83] Zengo, Zengo uncovers security vulnerabilities in popular Web3 transaction simulation solutions: The red pill attack - zengo, 2023, URL: https://zengo.com/zengo-uncovers-security-vulnerabilities-in-popular-web3-transaction-simulation-solutions-the-red-pill-attack/.
[84] Thodex, Exodus wallets vulnerable to sophisticated macos malware, 2023, URL: https://www.thodex.com/exodus-wallets-vulnerable-to-sophisticated-macos-malware/.
[85] Immunefi, Two novel crypto wallet exploits explained, 2022, URL: https://medium.com/immunefi/two-novel-crypto-wallet-exploits-explained-98e74e50d13f.
[86] CVE, CVE-2020-15302, 2020, URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-15302.
[87] CoinTelegraph, Slope wallets blamed for solana-based wallet attack, 2022, URL: https://cointelegraph.com/news/slope-wallets-blamed-for-solana-based-wallet-attack.
[88] Consensys, What's in a self-custody (non-custodial) wallet, anyway? 2021, URL: https://consensys.io/blog/whats-in-a-self-custody-non-custodial-wallet-anyway.
[89] Brave, Brave wallet vs. MetaMask - A comparison | brave, 2023, URL: https://brave.com/web3/difference-brave-wallet-metamask/.
[90] Ledger, On the security model of software wallets, 2021, URL: https://www.ledger.com/blog/software-wallets.
[91] V. Buterin, Y. Weiss, D. Tirosh, S. Nacson, A. Forshtat, K. Gazso, T. Hess, B. Vitalik, W. Yoav, T. Dror, N. Shahaf, F. Alex, G. Kristof, H. Tjaden, ERC-4337: Account abstraction using alt mempool, 2021, URL: https://eips.ethereum.org/EIPS/eip-4337.
[92] TON Developers, TON wallet-V4 smart contract specification, 2023, https://github.com/ton-blockchain/wallet-contracts. Defines the contract used by TON Space accounts.
[93] B.R. Desk, TON Blockchain Faces Criticism over Centralisation and Liquidity, BeInCrypto, 2025, URL: https://beincrypto.com/ton-centralisation-liquidity. Discusses TON validator centralisation and related risks.
[94] H. Security, Mobile WebView Risks in Telegram MiniApps, Technical Report, Halborn, 2024, URL: https://halborn.com/blog/telegram-miniapp-security. Analysis of generic WebView attack surfaces relevant to TON Space.
[95] S. Palladino, The parity wallet hack explained, 2017, July-2017. [Online]. Available: https://blog.zeppelin.solutions/on-the-parity-wallet-multisighack-405a8c12e8f7.
[96] C. Parisi, D. Budorin, O. Khalavka, Wallet security, 2023.
[97] Ledger, How can you sign online transactions when your private key is offline, 2022, URL: https://www.ledger.com/academy/how-can-you-sign-online-transactions-when-your-private-key-is-offline.
[98] CoinTelegraph, Ledger vulnerability put entire DApp ecosystem at risk: Finance redefined, 2023, URL: https://cointelegraph.com/news/ledger-vulnerability-put-entire-dapp-ecosystem-at-risk-finance-redefined.
[99] Ledger, Firmware 1.4: Deep dive into three vulnerabilities which have been fixed, 2018.
[100] Coin Desk, Security researchers break ledger wallets with simple antennae, 2018, URL: https://www.coindesk.com/markets/2018/12/28/security-researchers-break-ledger-wallets-with-simple-antennae.
[101] Freemindtronic, Ledger security breaches from 2017 to 2023: How to protect yourself from hackers, 2023, URL: https://freemindtronic.com/ledger-security-breaches-how-to-protect-your-cryptocurrencies.
[102] S. Akter, K. Khalil, M. Bayoumi, A survey on hardware security: Current trends and challenges, IEEE Access 11 (2023) 77543–77565, http://dx.doi.org/10.1109/ACCESS.2023.3288696.

[103] A. Hajdu, N. Ivaki, I. Kocsis, A. Klenik, L. Gonczy, N. Laranjeiro, H. Madeira, A. Pataricza, Using fault injection to assess blockchain systems in presence of faulty smart contracts, IEEE Access 8 (2020) 190760–190783.

[104] H. Wang, W. Liu, W. Cai, Y. Lu, C. Wan, Efficient attacks on strong PUFs via covariance and Boolean modeling, ACM Trans. Des. Autom. Electron. Syst. (2024) http://dx.doi.org/10.1145/3687469.

[105] F. Courbon, S. Skorobogatov, C. Woods, Reverse engineering flash EEPROM memories using scanning electron microscopy, in: International Conference on Smart Card Research and Advanced Applications, Vol. 10146 LNCS, Springer, Springer, Cham, 2016, pp. 57–72, http://dx.doi.org/10.1007/978-3-319-54669-8{_}4.

[106] K. Chalkias, P. Chatzigiannis, Y. Ji, Broken proofs of solvency in blockchain custodial wallets and exchanges, in: International Conference on Financial Cryptography and Data Security, Springer, 2022, pp. 106–117.

[107] C. Telegraph, News, 2024, URL: https://cointelegraph.com/news.

[108] Y. Erinle, Y. Feng, J. Xu, N. Vadgama, P. Tasca, Shared-custodial wallet for multi-party crypto-asset management, Futur. Internet 17 (1) (2024) 7.

[109] P. Das, A. Erwig, S. Faust, Shared-custodial password-authenticated deterministic wallets, 2024, pp. 338–359, http://dx.doi.org/10.1007/978-3-031-71073-5{_}16/FIGURES/4, URL: https://link.springer.com/chapter/10.1007/978-3-031-71073-5_16.

[110] Zengo, What is zengo's recovery kit?, 2024, URL: https://help.zengo.com/en/articles/2603673-what-is-zengo-s-recovery-kit.

[111] Ledger, Personal security device: The master seed, 2024, URL: https://developers.ledger.com/docs/device-app/architecture/psd/masterseed.

[112] V. Nair, D. Song, Multi-Factor Key Derivation Function (MFKDF) for Fast, Flexible, Secure, & Practical Key Management.

[113] V. Nair, D. Song, Decentralizing custodial wallets with MFKDF, in: 2023 IEEE International Conference on Blockchain and Cryptocurrency, ICBC, IEEE, 2023, pp. 1–9.

[114] G. Andresen, M-of-N standard transactions, 2011, URL: https://github.com/bitcoin/bips/blob/master/bip-0011.mediawiki.

[115] Y. Lindell, Secure multiparty computation, Commun. ACM 64 (1) (2020) 86–96.

[116] Web3 Auth, MPC architecture, 2024, URL: https://web3auth.io/docs/infrastructure/mpc-architecture.

[117] Binance, Embracing the future of Web3: Binance's innovative MPC wallet, 2023, URL: https://www.binance.com/en/square/post/1678388.

[118] Ethereum, ERC-1271: Standard signature validation method for contracts, 2018, URL: https://eips.ethereum.org/EIPS/eip-1271.

[119] Ethereum, ERC-6492: Signature validation for predeploy contracts, 2023, URL: https://eips.ethereum.org/EIPS/eip-6492.

[120] Vault12, What is BIP-39? Seed phrases, passphrases and security, 2025, URL: https://vault12.com/learn/crypto-security-basics/what-is-bip39/.

[121] T. Docs, Use of SLIP-39 in trezor core, 2024, URL: https://docs.trezor.io/trezor-firmware/core/misc/slip0039.html.

[122] SLIP-39 – Cryptocurrency seed generation, backup & recovery, 2024, URL: https://slip39.com/.

[123] Recovering a lost 12-word phrase with coinbase wallet cloud backup, 2024, URL: https://help.coinbase.com/en/wallet/other-topics/backups-and-recovery.

[124] V. Buterin, Why we need wide adoption of social recovery wallets, 2021, URL: https://vitalik.eth.limo/general/2021/01/11/recovery.html. (Accessed 12 August 2025).

[125] DeRec, 2023, https://github.com/derecalliance/protocol/blob/main/protocol.md.

[126] Recover your smart wallet. URL: https://help.coinbase.com/en/wallet/getting-started/smart-wallet-recovery.

[127] Argent, Off-chain recovery, 2021, URL: https://www.argent.xyz/blog/off-chain-recovery.

[128] Decrypt, FTX had 5 billion when it was supposed to have 20 billion, 2023, https://decrypt.co/202223/ftx-had-5-billion-when-it-was-supposed-to-have-20-billion.

[129] O. Yomtov, Fireblocks researchers uncover first account abstraction wallet vulnerability, 2023, URL: https://www.fireblocks.com/blog/fireblocks-researchers-uncover-first-account-abstraction-wallet-vulnerability/.

[130] Y. Hu, S. Wang, G.H. Tu, L. Xiao, T. Xie, X. Lei, C.Y. Li, Security threats from bitcoin wallet smartphone applications: Vulnerabilities, attacks, and countermeasures, in: CODASPY 2021 - Proceedings of the 11th ACM Conference on Data and Application Security and Privacy, Vol. 12, Association for Computing Machinery, Inc, 2021, pp. 89–100, http://dx.doi.org/10.1145/3422337.3447832/SUPPL{_}FILE/CODASPY21-FP331.MP4.

[131] D. He, S. Li, C. Li, S. Zhu, S. Chan, W. Min, N. Guizani, Security analysis of cryptocurrency wallets in android-based applications, IEEE Netw. 34 (6) (2020) 114–119, http://dx.doi.org/10.1109/MNET.011.2000025.

[132] SlowMist, Bybit's $1.5 billion theft unveiled: Safewallet front-end code tampered , 2025, URL: https://slowmist.medium.com/bybits-1-5-billion-theft-unveiled-safe-wallet-front-end-code-tampered-84b78f0fa9c2.

[133] Halborn, How the bybit hack happened and how to prevent the next one with seraph, 2025, URL: https://www.halborn.com/blog/post/how-the-bybit-hack-happened-and-how-to-prevent-the-next-one-with-seraph.

[134] M.S. Uddin, M. Mannan, A. Youssef, Horus: A security assessment framework for android crypto wallets, in: Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, Vol. 399 LNICST, Springer, Cham, 2021, pp. 120–139, http://dx.doi.org/10.1007/978-3-030-90022-9{_}7.

[135] E.O. Kiktenko, M.A. Kudinov, A.K. Fedorov, Detecting brute-force attacks on cryptocurrency wallets, in: Abramowicz Witold, R. Corchuelo (Eds.), Business Information Systems Workshops, Vol. 373 LNBIP, Springer International Publishing, Cham, 2019, pp. 232–242, http://dx.doi.org/10.1007/978-3-030-36691-9{_}20.

[136] A. Shaikh, Survey paper on security analysis of crypto-currency exchanges, Int. Res. J. Mod. Eng. Technol. Sci. 7 (2022).

[137] R. Rokhjavan, Securing multi-party crypto wallets, 2023, URL: https://DalSpace.library.dal.ca//handle/10222/82540.

[138] Y. Balakrishnan, P.N. Renjith, An analysis on keylogger attack and detection based on machine learning, in: 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering, ICECONF, IEEE, 2023, pp. 1–8.

[139] K. Weichbroth Pawełand Wereszko, H. Anacka, J. Kowal, P. Weichbroth, K. Wereszko, H. Anacka, J. Kowal, Security of cryptocurrencies: A view on the state-of-the-art research and current developments, Sensors 23 (6) (2023) 3155, http://dx.doi.org/10.3390/s23063155.

[140] S.S. Team, Malicious npm packages target TON wallet users, 2024, https://socket.dev/blog/malicious-npm-packages-target-ton-wallet. Initial disclosure of `@ton-wallet/create` clipboard-sniffer.

[141] G.S. Lab, Malicious TON package `ton-connect-sdk`—Security advisory, 2024, https://github.com/advisories/GHSA-xxxx. Details silent mnemonic exfiltration via post-install script.

[142] B. Staff, Clipboard-sniffing npm modules slip past security scans, 2024, https://beincrypto.com/npm-clipboard-sniffer-analysis. Follow-up on TON-related npm supply-chain attacks.

[143] SlowMist, Telegram phishing bot wave leads to batch wallet drains, 2025, URL: https://slowmist.com/telegram-bot-phishing-2025. June 2025 incident report.

[144] J. Breier, X. Hou, How practical are fault injection attacks, really? IEEE Access 10 (2022) 113122–113130, http://dx.doi.org/10.1109/ACCESS.2022.3217212.

[145] A. Robinson, C. Corcoran, J. Waldo, New risks in ransomware: supply chain attacks and cryptocurrency, Sci. Technol. Public Policy Program Rep. (2022) URL: https://dash.harvard.edu/handle/1/37373233.

[146] M.K. Shrivas, T.Y. Dean, S.S. Brunda, The disruptive blockchain security threats and threat categorization, in: 2020 First International Conference on Power, Control and Computing Technologies, ICPC2T, IEEE. Institute of Electrical and Electronics Engineers Inc, 2020, pp. 327–338, http://dx.doi.org/10.1109/ICPC2T48082.2020.9071475.

[147] M. Brengel, C. Rossow, Identifying key leakage of bitcoin users, in: Research in Attacks, Intrusions, and Defenses: 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings, Vol. 21, Springer, 2018, pp. 623–643.

[148] D. Park, M. Choi, G. Kim, D. Bae, H. Kim, S. Hong, Stealing keys from hardware wallets: A single trace side-channel attack on elliptic curve scalar multiplication without profiling, IEEE Access 11 (February) (2023) 44578–44589, http://dx.doi.org/10.1109/ACCESS.2023.3273150.

[149] P.C. Kocher, Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems, in: Advances in Cryptology—CRYPTO'96: 16th Annual International Cryptology Conference Santa Barbara, California, USA August 18–22, 1996 Proceedings 16, Vol. 1109, Springer, Springer, Berlin, Heidelberg, 1996, pp. 104–113, http://dx.doi.org/10.1007/3-540-68697-5{_}9.

[150] Zerocap, Ledger hack 2023, 2023, URL: https://zerocap.com/insights/snippets/ledger-hack-2023/.

[151] M.E. Ahmed, H. Kim, M. Park, Mitigating DNS query-based DDoS attacks with machine learning on software-defined networking, in: Proceedings - IEEE Military Communications Conference MILCOM 2017-October, 11–16, Institute of Electrical and Electronics Engineers Inc, 2017, http://dx.doi.org/10.1109/MILCOM.2017.8170802.

[152] S. Al-Mashhadi, S. Manickam, A brief review of blockchain-based DNS systems, Int. J. Internet Technol. Secur. Trans. 10 (4) (2020) 420–432, http://dx.doi.org/10.1504/IJITST.2020.108134.

[153] P. Ekparinya, V. Gramoli, G. Jourjon, Impact of man-in-the-middle attacks on ethereumekparinya2018impact, in: 2018 IEEE 37th Symposium on Reliable Distributed Systems, SRDS, IEEE, 2018, pp. 11–20.

[154] A. Pillai, V. Saraswat, A. VR, Smart wallets on blockchain—attacks and their costs, in: Smart City and Informatization: 7th International Conference, ISCI 2019, Guangzhou, China, November 12–15, 2019, Proceedings, Vol. 7, Springer, 2019, pp. 649–660.

[155] CoinBureau, Cryptocurrency exchange Ether Delta hacked in DNS hijacking scheme, 2021, URL: https://coinbureau.com/news/funds-stolen-etherdelta-suffers-dns-hack/.

[156] N. Mutual, How was MEW (MyEtherWallet) DNS spoofed? 2022, URL: https://neptunemutual.com/blog/how-was-mew-myetherwallet-dns-spoofed/.

[157] R. Chandan, R. St, V. Pallotti, R. Mahajan, R. Roychaudhary, Protective Mechanism form DDoS Attack for Cryptocoin, Technical Report, URL: https://www.researchgate.net/publication/353753938.

[158] K. Krombholz, H. Hobel, M. Huber, E. Weippl, Advanced social engineering attacks, J. Inf. Secur. Appl. 22 (2015) 113–122, http://dx.doi.org/10.1016/J.JISA.2014.09.005.
[159] J. Ferdous, R. Islam, A. Mahboubi, M.Z. Islam, A review of state-of-the-art malware attack trends and defense mechanisms, IEEE Access 11 (2023) 121118–121141, http://dx.doi.org/10.1109/ACCESS.2023.3328351.
[160] N. Ivanov, Q. Yan, Ethclipper: a clipboard meddling attack on hardware wallets with address verification evasion, in: 2021 IEEE Conference on Communications and Network Security, CNS, IEEE, 2021, pp. 191–199.
[161] M. Guri, Beatcoin: Leaking private keys from air-gapped cryptocurrency wallets, in: 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, 2018, pp. 1308–1316.
[162] CertiK, CertiK - 2022 year in review - crypto wallet security incidents, 2023, URL: https://www.certik.com/resources/blog/01iz10lvnaAIcuNZ2zNJqA-2022-year-in-review-crypto-wallet-security-incidents.
[163] Halborn, URL: https://www.halborn.com/blog/.
[164] CoinTelegraph, HTX exchange loses $13.6M in hot wallet hack: Report, 2023, URL: https://cointelegraph.com/news/htx-exchange-loses-14-million-hot-wallet-hack.
[165] Merkle Science, Hack track: Analysis on BitMart hack, 2021, URL: https://blog.merklescience.com/hacktrack/hack-track-analysis-on-bitmart-hack.
[166] DailyCoin, SlowMist flags escalated phishing attacks in the TON ecosystem, 2025, URL: https://dailycoin.com/slowmist-flags-escalated-phishing-attacks-in-the-ton-ecosystem/. (Accessed 01 July 2025).
[167] Halborn, MetaMask warns of new "address poisoning" crypto scam, 2023, URL: https://www.halborn.com/blog/post/metamask-warns-of-new-address-poisoning-crypto-scam.
[168] Bitcoin Magazine, Bitfloor hacked, $250,000 missing, 2012, URL: https://bitcoinmagazine.com/business/bitfloor-hacked-250000-missing-1346821046/.
[169] Bitcoin Magazine, Exchange site Bitcoinica hacked, US$90,000 stolen, 2012, URL: https://bitcoinmagazine.com/business/the-bitcoinica-linode-theft-and-what-it-means-for-bitcoin-1330805009.
[170] BitcoinNewsLooking Back At LCX Hack From January 2022, Looting of the fox: The story of sabotage at ShapeShift, 2016, URL: https://news.bitcoin.com/looting-fox-sabotage-shapeshift/.
[171] BlockOnomi, Exit scam or hack? Canada's MapleChange loses $5 million of customer funds, 2018, URL: https://blockonomi.com/maplechange-hacked-scam/.
[172] CoinDesk, URL: https://www.coindesk.com.
[173] CoinDesk, Poloniex loses 12.3% of its bitcoins in latest bitcoin exchange hack, 2014, URL: https://www.coindesk.com/markets/2014/03/05/poloniex-loses-123-of-its-bitcoins-in-latest-bitcoin-exchange-hack/.
[174] CoinDesk, Details of $5 million bitstamp hack revealed, 2015, URL: https://www.coindesk.com/markets/2015/07/01/details-of-5-million-bitstamp-hack-revealed/.
[175] CoinDesk, Electrum wallet attack may have stolen as much as 245 Bitcoin, 2018, URL: https://www.coindesk.com/markets/2018/12/28/electrum-wallet-attack-may-have-stolen-as-much-as-245-bitcoin.
[176] CoinDesk, Czech bitcoin exchange Bitcash.cz hacked - 4,000 user wallets emptied, 2021, URL: https://www.coindesk.com/markets/2013/11/12/czech-bitcoin-exchange-bitcashcz-hacked-and-up-to-4000-user-wallets-emptied/.
[177] CoinDesk, Crypto exchange deribit loses $28M in hot wallet hack, pauses withdrawals, 2022, URL: https://www.coindesk.com/business/2022/11/02/crypto-exchange-deribit-loses-28m-in-hot-wallet-hack/.
[178] CoinDesk, FTX news coverage: FTX hack or inside job? Blockchain experts examine clues and a 'stupid mistake', 2022, URL: https://www.coindesk.com/business/2022/11/14/ftx-hack-or-inside-job-blockchain-experts-examine-clues-and-a-stupid-mistake/.
[179] CoinDesk, Indonesian crypto exchange indodax hacked for $22M in ether (ETH), Bitcoin (BTC), tron (TRX), 2024, URL: https://www.coindesk.com/markets/2024/09/11/indonesian-crypto-exchange-indodax-hacked-for-22m-pauses-activity-before-bigger-hit/.
[180] CoinMarketCap, 'Coincheck hack - one of the biggest crypto hacks in history, 2018, URL: https://coinmarketcap.com/academy/article/coincheck-hack-one-of-the-biggest-crypto-hacks-in-history/.
[181] CoinMarketCap, BitGrail hack - one of the largest crypto hacks in history, 2023, URL: https://coinmarketcap.com/academy/article/bitgrail-hack-one-of-the-largest-crypto-hacks-in-history.
[182] CoinTelegraph, BlackWallet hack: $400K in stellar stolen, hosting provider possibly at fault, 2018, URL: https://cointelegraph.com/news/blackwallet-hack-400k-in-stellar-stolen-hosting-provider-possibly-at-fault.
[183] CoinTelegraph, Japanese crypto exchange bitpoint suffers $32 million hack?, 2019, URL: https://cointelegraph.com/news/japanese-crypto-exchange-bitpoint-suffers-32-million-hack.
[184] CoinTelegraph, Monero's community wallet loses all funds after attack, 2023, URL: https://cointelegraph.com/news/monero-community-wallet-loses-all-funds-after-attack/.
[185] CryptoSlate, ZachXBT calls out gate.io for keeping 2018 hack under wraps, 2018, URL: https://cryptoslate.com/zachxbt-calls-out-gate-io-for-keeping-2018-hack-under-wraps/.
[186] CryptoSlate, Ripple co-founder chris larsen's XRP wallet hacked for estimated $112 million, 2024, URL: https://cryptoslate.com/ripple-co-founder-chris-larsens-xrp-wallet-hacked-for-estimated-112-million/.
[187] N. De, Hackers steal $40.7 million in Bitcoin from crypto exchange binance, 2021, URL: https://www.coindesk.com/markets/2019/05/07/hackers-steal-407-million-in-bitcoin-from-crypto-exchange-binance/.
[188] Decrypt, URL: https://decrypt.co.
[189] Decrypt, Hacker robs crypto exchange bitrue of $23M in ethereum, SHIB, other assets, 2023, URL: https://decrypt.co/136751/hacker-robs-bitrue-23m-ethereum-shib-assets/.
[190] Hacken, How to avoid a hack: Cryptopia 'success' case , 2019, URL: https://hacken.io/discover/how-to-avoid-a-hack-cryptopia-success-case/.
[191] Hacken, Looking back at LCX hack from january 2022, 2022, URL: https://hacken.io/industry-news-and-insights/lcx-hack-january-2022/.
[192] IOTA, MoonPay integration is responsible for hack of the trinity wallet, 2020, URL: https://www.crypto-news-flash.com/iota-moonpay-integration-responsible-for-hack/.
[193] Ledger, Remembering the mintpal hack - October 2014 $3.500.000 loss in crypto assets, 2018, URL: https://www.ledger.com/remembering-the-mintpal-hack.
[194] Merkle Science, Hack track: Analysis of liquid global security breach, 2021, URL: https://blog.merklescience.com/hacktrack/hack-track-initial-analysis-of-liquid-global-security-breach.
[195] N. Mutual, Analysis of the fantom foundation exploit | by neptune mutual | neptune mutual | medium, 2023, URL: https://medium.com/neptune-mutual/analysis-of-the-fantom-foundation-exploit-3990ba8f6d94.
[196] SlowMist, The real cause of the wintermute exploit | by SlowMist | medium, 2022, URL: https://slowmist.medium.com/the-real-cause-of-the-wintermute-exploit-10da7e404b3b.
[197] The Block, VinDAX got hacked; lost 'half a million USD' worth of tokens, 2019, URL: https://www.theblock.co/post/46408/little-known-asian-crypto-exchange-vindax-got-hacked-lost-half-a-million-usd-worth-of-tokens.
[198] Thomas Silkjær, Overview of the "gatehub hack". On june 1 we were made aware of a theft…, 2019, URL: https://medium.com/xrp-forensics/overview-of-the-gatehub-hack-f88a441c9203.
[199] ZDNET, Altsbit plans exit after hack leaves cryptocurrency exchange out of pocket, 2020, https://www.zdnet.com/article/altsbit-says-hack-has-left-the-cryptocurrency-exchange-with-next-to-no-funds/. [Online; (Accessed 10 February 2020)].
[200] Rekt, News. URL: https://rekt.news/.
[201] CoinTelegram, Crypto exchange upbit confirms theft of 342,000 ether — $50M, 2019, URL: https://cointelegraph.com/news/breaking-crypto-exchange-upbit-confirms-theft-of-342-000-ether-50m.
[202] P. Praitheeshan, L. Pan, J. Yu, J. Liu, R. Doss, Security analysis methods on ethereum smart contract vulnerabilities: a survey, 2019, arXiv preprint arXiv:1908.08605.
[203] J. Galbally, S. Marcel, J. Fierrez, Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition, IEEE Trans. Image Process. 23 (2) (2013) 710–724.
[204] M. Kim, J. Suh, H. Kwon, A study of the emerging trends in SIM swapping crime and effective countermeasures, in: Proceedings - 2022 IEEE/ACIS 7th International Conference on Big Data, Cloud Computing, and Data Science, BCD 2022, Institute of Electrical and Electronics Engineers Inc, 2022, pp. 240–245, http://dx.doi.org/10.1109/BCD54882.2022.9900510.
[205] H. Altuwaijri, S. Ghouzali, Android data storage security: A review, J. King Saud Univ. Comput. Inf. Sci. 32 (5) (2020) 543–552.
[206] M. Garcia-Bosque, G. Diez-Senorans, C. Sanchez-Azqueta, S. Celma, Introduction to physically unclonable fuctions: Properties and applications, in: ECCTD 2020 - 24th IEEE European Conference on Circuit Theory and Design, Institute of Electrical and Electronics Engineers Inc, 2020, http://dx.doi.org/10.1109/ECCTD49232.2020.9218404.
[207] D. Park, J. Kim, H.S. Kim, S. Hong, Cloning hardware wallet without valid credentials through side-channel analysis of hash function, IEEE Access (2024) http://dx.doi.org/10.1109/ACCESS.2024.3440370.
[208] J.S. Ko, J. Kwak, Private key recovery on Bitcoin with duplicated signatures, KSII Trans. Internet Inf. Syst. (TIIS) 14 (3) (2020) 1280–1300, http://dx.doi.org/10.3837/TIIS.2020.03.020.
[209] A. Kapoor, A. Gupta, R. Gupta, S. Tanwar, G. Sharma, I.E. Davidson, Ransomware detection, avoidance, and mitigation scheme: a review and future directions, Sustainability 14 (1) (2021) 8, http://dx.doi.org/10.3390/SU14010008.
[210] T. Girdler, V.G. Vassilakis, Implementing an intrusion detection and prevention system using software-defined networking: Defending against ARP spoofing attacks and blacklisted MAC addresses, Comput. Electr. Eng. 90 (2021) 106990, http://dx.doi.org/10.1016/J.COMPELECENG.2021.106990.
[211] A. Zimba, Z. Wang, M. Mulenga, Cryptojacking injection: A paradigm shift to cryptocurrency-based web-centric internet attacks, J. Org. Comput. Electron. Commer. 29 (1) (2019) 40–59.

[212] H. Aldawood, G. Skinner, An advanced taxonomy for social engineering attacks, Int. J. Comput. Appl. 177 (30) (2020) 1–11.

[213] A. Aratani, A. Kanai, Authentication method against shoulder-surfing attacks using secondary channel, in: 2015 IEEE International Conference on Consumer Electronics, ICCE 2015, Institute of Electrical and Electronics Engineers Inc, 2015, pp. 430–431, http://dx.doi.org/10.1109/ICCE.2015.7066474.

[214] S.G. Bhirud, V. Katkar, Light weight approach for IP-ARP spoofing detection and prevention, Asian Himal. Int. Conf. Internet (2011) http://dx.doi.org/10.1109/AHICI.2011.6113951.

[215] H. Byun, J. Kim, Y. Jeong, B. Seok, S. Gong, C. Lee, A security analysis of cryptocurrency wallets against password brute-force attacks, Electron. 13 (13) (2024) 2433, http://dx.doi.org/10.3390/ELECTRONICS13132433, 2024, 13 2433.

[216] R. Chaganti, B. Bhushan, V. Ravi, The role of blockchain in DDoS attacks mitigation: techniques, open challenges and future directions, 2022, arXiv preprint arXiv:2202.03617.

[217] G. Destefanis, M. Marchesi, M. Ortu, R. Tonelli, A. Bracciali, R. Hierons, Smart contracts vulnerabilities: A call for blockchain software engineering? in: 2018 IEEE 1st International Workshop on Blockchain Oriented Software Engineering, IWBOSE 2018 - Proceedings 2018-January, Institute of Electrical and Electronics Engineers Inc, 2018, pp. 19–25, http://dx.doi.org/10.1109/IWBOSE.2018.8327567.

[218] Fireblocks, Manage destination addresses, 2025, URL: https://developers.fireblocks.com/docs/whitelist-addresses.

[219] H. Gupta, S. Mondal, R. Majumdar, N.S. Ghosh, S. Suvra Khan, N.E. Kwanyu, V.P. Mishra, Impact of side channel attack in information security, in: Proceedings of 2019 International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2019, Institute of Electrical and Electronics Engineers Inc, 2019, pp. 291–295, http://dx.doi.org/10.1109/ICCIKE47802.2019.9004435.

[220] W. Hu, C.-H.H. Chang, A. Sengupta, S. Bhunia, R. Kastner, H. Li, An overview of hardware security and trust: Threats, countermeasures, and design tools, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 40 (6) (2020) 1010–1038, http://dx.doi.org/10.1109/TCAD.2020.3047976.

[221] C.Y. Kim, K. Lee, Risk management to cryptocurrency exchange and investors: Guidelines to prevent potential threats, in: 2018 International Conference on Platform Technology and Service, PlatCon 2018, Institute of Electrical and Electronics Engineers Inc, 2018, http://dx.doi.org/10.1109/PLATCON.2018.8472760.

[222] A.M. Shuvo, T. Zhang, F. Farahmandi, M. Tehranipoor, A comprehensive survey on non-invasive fault injection attacks, Cryptol. EPrint Arch. (2023).

[223] D. Tymokhanov, O. Shlomovits, D. Tymokhanov Velas, O. Shlomovits ZenGo-X, Alpha-rays: Key extraction attacks on threshold ecdsa implementations, Cryptol. EPrint Arch. (2021).

[224] M. Cai, Z. Wu, J. Zhang, Research and prevention of rogue AP based MitM in wireless network, in: Proceedings - 2014 9th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC 2014, Institute of Electrical and Electronics Engineers Inc, 2014, pp. 538–542, http://dx.doi.org/10.1109/3PGCIC.2014.105.

[225] Y. Liu, M. Dong, K. Ota, J. Li, J. Wu, Deep reinforcement learning based smart mitigation of DDoS flooding in software-defined networks, in: 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD, IEEE, 2018, pp. 1–6.

[226] S. Sathwara, C. Parekh, Distributed denial of service attacks–TCP syn flooding attack mitigation, Int. J. Adv. Res. Comput. Sci. 8 (5) (2017).

[227] Indusface, How to implement root detection in android applications?. https://www.indusface.com/learning/how-to-implement-root-detection-in-android-applications/.

[228] Z. Qi, B. Li, Q. Lin, M. Yu, M. Xia, H. Guan, SPAD: Software protection through anti-debugging using hardware-assisted virtualization, J. Inf. Sci. Eng. 28 (5) (2012) 813–827.

[229] A. Rengarajan, R. Sugumar, C. Jayakumar, Secure verification technique for defending IP spoofing attacks, Int. Arab. J. Inf. Technol. 13 (2) (2016) 302–309.

[230] R.U. Khan, X. Zhang, R. Kumar, A. Sharif, N.A. Golilarz, M. Alazab, An adaptive multi-layer botnet detection technique using machine learning classifiers, Appl. Sci. 9 (11) (2019) 2375.

[231] ENS, Ethereum name service, 2024, URL: https://ens.domains.

[232] P. Xia, H. Wang, Z. Yu, X. Liu, X. Luo, G. Xu, G. Tyson, Challenges in decentralized name management: The case of ENS, in: Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC, Association for Computing Machinery, 2022, pp. 65–82, http://dx.doi.org/10.1145/3517745.3561469/SUPPL{_}FILE/741.M4V.

[233] G. Creech, New approach to return-oriented programming exploitation mitigation, Inf. Secur. J. Glob. Perspect. 26 (3) (2017) 105–120, http://dx.doi.org/10.1080/19393555.2017.1308583.

[234] X. Wang, On the feasibility of detecting software supply chain attacks, in: Proceedings - IEEE Military Communications Conference MILCOM 2021-November, Institute of Electrical and Electronics Engineers Inc, 2021, pp. 458–463, http://dx.doi.org/10.1109/MILCOM52596.2021.9652901.

[235] FIDO Alliance, How FIDO works. URL: https://fidoalliance.org/fido2/.

[236] A. Tereshkin, Evil maid goes after PGP whole disk encryption, 2010, p. 2, http://dx.doi.org/10.1145/1854099.1854103, URL: https://dl.acm.org/doi/10.1145/1854099.1854103.

[237] H. Seol, M. Kim, T. Kim, Y. Kim, L.-S.S. Kim, Amnesiac DRAM: A proactive defense mechanism against cold boot attacks, IEEE Trans. Comput. 70 (4) (2019) 539–551, http://dx.doi.org/10.1109/TC.2019.2946365.

[238] Bitcoin Magazine, Bitcoin hardware wallet review ledger may have caught up to trezor with nano S, 2016, URL: https://bitcoinmagazine.com/technical/bitcoin-hardware-wallet-review-ledger-may-have-caught-up-to-trezor-with-nano-s-1473944111.

[239] X. Lou, T. Zhang, J. Jiang, Y. Zhang, A survey of microarchitectural side-channel vulnerabilities, attacks, and defenses in cryptography, ACM Comput. Surv. 54 (6) (2021) http://dx.doi.org/10.1145/3456629.

[240] U. Ali, S.A.R. Sahni, O. Khan, Characterization of timing-based software side-channel attacks and mitigations on network-on-chip hardware, ACM J. Emerg. Technol. Comput. Syst. 19 (3) (2023) http://dx.doi.org/10.1145/3585519/ASSET/4B39CFE6-E67D-4331-B538-A7FC16C1EA09/ASSETS/GRAPHIC/JETC-2022-0036-F19.JPG.

[241] F. Mosquera, K. Kavi, G. Mehta, L.K. John, Guard cache: Creating false cache hits and misses to mitigate side-channel attacks, in: 2023 Silicon Valley Cybersecurity Conference, SVCC 2023, Institute of Electrical and Electronics Engineers Inc, 2023, http://dx.doi.org/10.1109/SVCC56964.2023.10165527.

[242] SecuX, Technical analysis report: Bybit exchange security breach - principles, process, and recommendations , 2025, URL: https://secuxtech.com/blogs/blog/technical-analysis-report-bybit-exchange-security-breach-principles-process-and-recommendations?srsltid=AfmBOoo-RIqWVsuKPfq94cmiUrF9TvNCcvUNJ3jhsar0OPC14Y4ztk6P.

[243] Cyfrin, The safe wallet hack that led to bybit's $1.4b heist , 2025, URL: https://www.cyfrin.io/blog/safe-wallet-hack-bybit-exploit.

[244] C. Intelligence, Blockchain auditors say $4M crypto theft enabled by logging tech, 2022, URL: https://www.cybersecurityintelligence.com/blog/blockchain-auditors-say-4m-crypto-theft-enabled-by-logging-tech-6463.html.

[245] C. Jones, Slope wallet breach caused by misconfigured logging, 2022, URL: https://www.itpro.com/security/2022/slope-wallet-breach-logging.

[246] D. Team, Slope non-custodial wallet hack: Key facts & timeline, 2022, URL: https://dailycoin.com/slope-denies-evidence-of-all-security-layers-being-compromised.

[247] OtterSec, OtterSec security audit: Slope wallet forensics, 2022, URL: https://ottersec.com/reports/slope-wallet-forensics.

[248] Z. Security, Zellic analysis: Root cause of slope wallet breach, 2022, URL: https://www.zellic.io/blog/slope-wallet-root-cause-analysis.

[249] Slope Finance, Official statement on 2022 security incident, 2022, URL: https://slope.finance/blog/official-slope-incident-statement.

[250] Immunefi, Immunefi bug bounty and vulnerability disclosure platform, 2024, URL: https://immunefi.com/responsible-publication/.

[251] IC3, Initiative for cryptocurrencies and contracts (IC3), 2024, URL: https://www.initc3.org.

[252] Stanford University, Stanford center for blockchain research (CBR), 2024, URL: https://cbr.stanford.edu.

[253] SlowMist, Slowmist hacked - blockchain incident database, 2024, URL: https://hacked.slowmist.io.

[254] X. Feng, Q. Li, K. Sun, Man-in-the-middle attacks without rogue AP: when WPAs meet ICMP redirects, in: X. Feng, Q. Li, K. Sun, Y. Yang, K. Xu (Eds.), 2023 IEEE Symposium on Security and Privacy, SP, Ieeexplore.Ieee.Org, 2023, http://dx.doi.org/10.1109/SP46215.2023.10179441, 2023•ieeexplore.Ieee.Org.

[255] V. Tirronen, Stopping injection attacks with code and structured data, Intell. Syst. Control Autom. Sci. Eng. 93 (2018) 219–231, http://dx.doi.org/10.1007/978-3-319-75307-2{_}13.

[256] Y. Feng, J. Xu, L. Weymouth, University blockchain research initiative (UBRI): Boosting blockchain education and research, IEEE Potentials 41 (6) (2022) 19–25.