

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

From Data Behavior to Code Analysis: A Multimodal Study on Security and Privacy Challenges in Blockchain-Based DApp

HAOYANG SUN* and YISHUN WANG*, Hainan University, China
XIAOQI LI, Hainan University, China

The recent proliferation of blockchain-based decentralized applications (DApp) has catalyzed transformative advancements in distributed systems, with extensive deployments observed across financial, entertainment, media, and cybersecurity domains. These trustless architectures, characterized by their decentralized nature and elimination of third-party intermediaries, have garnered substantial institutional attention. Consequently, the escalating security challenges confronting DApp demand rigorous scholarly investigation. This study initiates with a systematic analysis of behavioral patterns derived from empirical DApp datasets, establishing foundational insights for subsequent methodological developments. The principal security vulnerabilities in Ethereum-based smart contracts developed via Solidity are then critically examined. Specifically, reentrancy vulnerability attacks are addressed by formally representing contract logic using highly expressive code fragments. This enables precise source code-level detection via bidirectional long short-term memory networks with attention mechanisms (BLSTM-ATT). Regarding privacy preservation challenges, contemporary solutions are evaluated through dual analytical lenses: identity privacy preservation and transaction anonymity enhancement, while proposing future research trajectories in cryptographic obfuscation techniques.

Additional Key Words and Phrases: Blockchain; Decentralized Applications; Privacy Protection; Smart Contracts; Deep Learning

1 INTRODUCTION

The emergence of Bitcoin has catalyzed unprecedented advancements in blockchain technology, positioning it as a pivotal research frontier. This technological evolution has undergone three distinct evolutionary phases: 1) The Blockchain 1.0 era, epitomized by Bitcoin, established the cryptocurrency paradigm through decentralized monetary systems and payment mechanisms, albeit with limited industrial applications. 2) The Blockchain 2.0 phase, marked by Ethereum's introduction of smart contracts with Turing-complete scripting capabilities, witnessed expanded application scenarios across banking, insurance, securities, and trust sectors through sophisticated development ecosystems. 3) The current DApp epoch, where blockchain serves as foundational infrastructure enabling cross-industry credential authentication, while DApp function as value-transfer vectors through cross-industry value transfer protocols.

Blockchain technology, frequently hailed as the linchpin of the Fourth Industrial Revolution, derives its transformative potential primarily through DApp implementations – architecturally defined as distributed applications underpinned by blockchain consensus mechanisms and self-executing smart contracts. As the hallmark of Blockchain 3.0, DApp ecosystems critically influence the maturation trajectory of blockchain infrastructures. Crucially, DApp are projected to epitomize future blockchain-enabled socioeconomic frameworks, particularly through their capacity to mediate trustless transaction frameworks across decentralized autonomous organizations.

Although the concept of blockchain was proposed by Satoshi Nakamoto [29] (or a team) as early as 2008, the application of this technology in the real world has only been in use for a few years, and there are still a series of issues that cannot be ignored, such as privacy protection and controllable supervision, the inability to achieve decentralization, security, and scalability simultaneously, and the operational efficiency of blockchain itself. According to data from the blockchain security

*The two authors contributed equally to this research.

Authors' addresses: Haoyang Sun; Yishun Wang, yishunwang@hainanu.edu.cn, Hainan University, Haikou, China; Xiaoqi Li, Hainan University, Haikou, China, csxqli@ieee.org.

company PeckShield (Pai Dun), there were 177 blockchain security incidents in 2019, resulting in economic losses as high as 7.679 billion US dollars, an increase of about 60% compared to 2018 [36]. According to incomplete statistics from the National Blockchain Vulnerability Database, the number of blockchain security incidents in 2020 reached 555, an increase of nearly 240% compared to 2019, with economic losses amounting to 17.9 billion US dollars [2]. According to data released by the foreign research institution Chainalysis, the amount of cryptocurrency crime in 2021 reached 14 billion US dollars, an increase of 79% year-on-year. The losses caused by cryptocurrency fraud cases reached 7.8 billion US dollars, an increase of 82% year-on-year, and the losses from hacking theft cases were approximately 3.2 billion US dollars, an increase of 516% year-on-year [15].

This paper will, based on existing research, collect data related to DApp, and combine the structural characteristics of blockchain to analyze the privacy protection and security of decentralized applications (DApp) on blockchain from two aspects: security and privacy protection.

The principal theoretical and methodological contributions of this paper are tripartite:

- First, a multidimensional analytical framework is developed for deconstructing DApp' behavioral patterns through heterogeneous data fusion techniques.
- Second, an innovative vulnerability detection paradigm is established by implementing Bidirectional Long Short-Term Memory networks with Attention mechanisms (BLSTM-ATT) to identify reentrancy vulnerabilities in Solidity-based smart contracts at source code granularity.
- Third, a systematic theoretical framework for privacy preservation is formulated through dual-aspect analysis of identity anonymization protocols and transaction obfuscation mechanisms [35], incorporating formal verification of zk-SNARKs implementations and quantitative assessment of differential privacy parameters.

1.1 Related Work

DApp, which are derived from underlying blockchain platforms, are decentralized applications running on top of smart contracts based on P2P peer-to-peer networks. The underlying blockchain technology provides them with trustworthy data recording. Through searching for relevant information on the network, it is found that research on DApp by related platforms mainly focuses on aspects such as application domains, application platforms, quality assessment, and DApp architectures [9, 11, 25, 27, 38], with corresponding data and behavior analyses. Research on the privacy protection and security of DApp is concentrated on the security and privacy protection of the underlying technology they employ, namely blockchain technology.

In recent years, research on the security and privacy protection of blockchain technology has been conducted. Bu et al. [3] investigated the security risks of blockchain systems, reviewed attack cases on blockchain systems, and analyzed the exploited vulnerabilities. Taylor P. J. et al. [45] conducted a systematic analysis of common blockchain security protocols. Singh S. et al.[44] provided a detailed analysis of potential blockchain security attacks and proposed existing solutions to these attacks. In addition, some researchers have offered insightful perspectives on vulnerability detection in smart contracts[4, 17, 18, 23].

Regarding privacy protection, Abdikhakimov and Islombek [1] introduced privacy protection mechanisms from three aspects: the blockchain network layer, the transaction layer, and the application layer. Zhang et al.[51] classified blockchain privacy protection technologies into address obfuscation, information hiding, and channel isolation and analyzed and compared the implementations of these three types of privacy protection technologies. Shen et al.[43] categorized blockchain privacy into identity privacy and transaction privacy and analyzed the security issues associated with these two types of privacy.

In terms of smart contract security issues, Christof Ferreira Torres et al. [47] proposed a hybrid simulator called CONFUZZIUS, which effectively identifies more bugs through constraint-solving and dynamic data analysis. Rao et al.[37] proposed a transaction-based classification detection method for Ethereum smart contracts. Palina Tolmach et al.[46] proposed formal models and specifications for smart contracts, as well as methods for verifying such specifications.

2 BACKGROUND

2.1 Blockchain

Blockchain is a decentralized distributed database composed of multiple interconnected blocks linked through cryptographic algorithms. Each block contains the hash value of the preceding block, transaction data, timestamps, and other relevant information[28]. By leveraging consensus algorithms, the blockchain network achieves a consensus mechanism, ensuring data synchronization and eliminating the possibility of data forgery by any single node, thereby enabling a trustless system. Through mutually agreed protocols and smart contracts, nodes interact and compete autonomously, ensuring the system operates independently without human intervention. Cryptographic algorithms enable any participant to query data records via public interfaces while preventing data modification or repudiation[22]. The chained structure facilitates efficient and rapid retrieval of transaction data, ensuring the traceability of data and transactions.

2.2 DApp

A DApp[10] represents the integration of traditional applications (APPs) with blockchain technology, typically operating on a peer-to-peer (P2P) network. It serves as an enhancement and extension of conventional applications, with the key distinction being its decentralized nature. In DApp, participant information is either anonymous or protected, and operations are conducted through nodes on a peer-to-peer network. Smart contracts provide the foundational framework for decentralization, enabling trustless interactions between participants. From a practical perspective, a DApp can be succinctly conceptualized as a combination of smart contracts and traditional applications[24], where smart contracts establish the prerequisites for decentralization. Structurally, DApp involves interactions between a front-end interface and users, as well as between smart contracts and the blockchain. This makes DApp publicly accessible programs that operate transparently on a network, leveraging blockchain's inherent properties of immutability and traceability.

2.3 Smart Contract

The concept of smart contracts[49] was first introduced by Nick Szabo in 1994. He defined a smart contract as "a set of promises, specified in digital form, including the protocols within which the parties perform on these promises." Smart contracts operate on distributed ledgers and can execute, verify, and enforce complex behaviors of distributed nodes based on predefined rules, without the need for third parties, thereby achieving functions such as programming and information exchange.

Smart contracts are intelligent electronic contracts that transform contractual agreements into code, which is then deployed on a blockchain. Once deployed, the code is publicly accessible and immutable. When external conditions change, such as a breach or contract expiration, smart contracts automatically trigger the execution of predefined actions. This automation ensures transparency, efficiency, and trust in transactions, eliminating the need for intermediaries and reducing the potential for disputes.

2.4 Ethereum

Ethereum[12] is a decentralized, open-source public blockchain platform with smart contract functionality. It features an integrated Turing-complete programming language, enabling users to develop decentralized applications (DApp) according to their specific requirements. As an application runtime platform, Ethereum ensures data transparency by making all data publicly accessible to nodes and immutable to third-party modifications. The Ethereum Virtual Machine (EVM) facilitates the execution and invocation of smart contracts. Additionally, Ethereum employs an account model, which reduces the cost of batch transaction processing, simplifies programming and development, and broadens the scope of application scenarios.

2.5 Security Threats

As an emerging application, DApp are increasingly recognized by enterprises and organizations for their underlying blockchain technology[26], which features decentralization, tamper resistance, and traceability. However, DApp still face significant security threats. In this context, we analyze the security threats of DApp from two perspectives: smart contract security and privacy protection.

2.5.1 Smart Contract Security.

Smart contract security is a critical component of DApp security. In 2016, vulnerabilities in the smart contracts of The DAO project led to the transfer of over 3.6 million Ether, resulting in losses exceeding USD 50 million[5]. This incident caused a temporary downturn in blockchain development.

2.5.2 Privacy Protection.

In blockchain systems, privacy protection primarily focuses on identity and transaction information, which can be divided into identity privacy protection and transaction privacy protection. According to a report by The Record in April 2021, over 533 million Facebook users' personal information was leaked on a hacking forum [34]. In June 2021, more than 700 million LinkedIn user data records were sold on a dark web platform. These incidents have drawn urgent attention to data security and privacy protection issues[32].

3 METHODOLOGY

3.1 Data Analytics for DApp

The advantages of DApp stem from the underlying blockchain technology, which enables data ownership and value transfer. DApp facilitate inter-industry integration, ensure product controllability and traceability, reduce operational and development costs, enhance transaction security, and improve user experience. Due to their decentralized nature, DApp are increasingly valued and adopted by enterprises and organizations. However, security and privacy protection issues in blockchain technology and smart contracts may introduce new challenges to DApp security. We have collected the behavioral data of DApp and analyzed their distribution from the perspectives of platform, type, and smart contract. This analysis of DApp data behavior forms the basis for our subsequent analysis.

3.1.1 Data Collection.

Data collection is essential for this study on DAPP security and privacy protection. This section introduces the dataset. We gathered DAPP-related data, including the number, types, platforms, and transactions, from four websites: State of the DApp, DAppReview, Top Blockchain DApp, and DApp.com, covering the period from April 2015 to February 2022. The data are presented in Table.1, and we performed statistical analysis of DAPP data behavior accordingly.

Platform	DApp Count	Smart Contract Count
Ethereum	2935	4890
Klaytn	80	316
EOS	331	550
<i>Steem</i>	79	177
Hive	56	105
POA	21	51
<i>xDai</i>	21	58
Neo	24	30
Obyte	17	162
OST	2	3
Loom	14	33
GoChain	7	17
<i>Blockstack</i>	24	0
TRON	88	281
ICON	16	36
NEAR	23	21
BSC	189	354
<i>Moonriver</i>	37	82

Table 1. Platform DApp and Smart Contract Counts

We collected data on 3,964 DApp. Based on their application scenarios, we categorized these DApp into 21 classes, including gaming, gambling, and finance. The distribution of these categories is shown in Table.2.

Category	DAPP Count
Games	680
Gambling	611
Social	415
Finance	386
Exchanges	264
Development	222
NFT	201
Def'i	192
Media	169
Wallet	138
Marketplaces	131

Category	DAPP Count
Governance	94
Security	87
Yield-farming	80
Property	81
Tools	61
Identity	48
Energy	34
Health	32
Insurance	20
Storage	18

Table 2. DAPP Category Distribution

In June 2018, EOS emerged as a new blockchain framework and gained attention for its scalability in decentralized applications. However, Ethereum's greater decentralization and longer application release history have led to the majority of DApp still being deployed on Ethereum. Therefore, we have collected and analyzed data on transactions, active users, and Ether from January 2018 to September 2020. Specific activity details are shown in Table.3.

	Transaction Volume	Active Users	ETH
Games	14734372	764798	300356
Gambling	9155186	466603	5037386
Social	667250	143384	6265
Finance	8415631	2091082	24311498
Exchanges	21377239	2153735	16068296
Development	1306938	311914	293079
Media	684475	231868	3300
Wallet	6990099	1395655	1004362
Market	2435308	190737	147025
Governance	330860	103686	1919
Security	2397611	773461	16899
Property	913607	106301	42736
Identity	353166	58768	4617
Energy	12870	7414	3237
Health	263	96	0
Insurance	5745	1969	0
Storage	1281920	574544	8
High Risk	4842315	1760754	8446005

Table 3. Ethereum Activity Data

3.2 Analysis Results

Based on the data collected in Section 3.1, we performed preprocessing and conducted a multi-angle analysis of DApp data behavior. This analysis addresses key questions, such as the current state of DApp and which types are more investment-worthy. The findings enhance our understanding of blockchain and provide a data foundation for security analysis and privacy protection.

3.2.1 Platform Distribution Data.

With the continuous development and improvement of blockchain technology, DApp have also been growing rapidly. The growing recognition of DApp's value has drawn increasing attention from organizations and enterprises. It should be noted that the quantity changes mentioned in the figure below refer to the current number of DApp, which is the difference between newly added and discontinued DApp.

Currently, the leading blockchain development platforms for DApp are Ethereum, EOS, and TRON. As shown in Figure.1, Ethereum remains the dominant platform. While its longer existence contributes to this dominance, the data also reflect Ethereum's advantages in DApp development, with a remarkable share of 74.04%.

Surveys indicate that the number of DApp deployed on Ethereum has consistently grown. Yet, the growth rate has shifted over time. From April 2015 to November 2018, the growth rate increased, but it has declined since November 2018. This change is mainly due to two factors: 1) Ethereum's security vulnerabilities, such as those in The DAO and Parity, which have raised risk concerns; and 2) the diversification of DApp development platforms, with the emergence of new platforms like EOS and TRON.

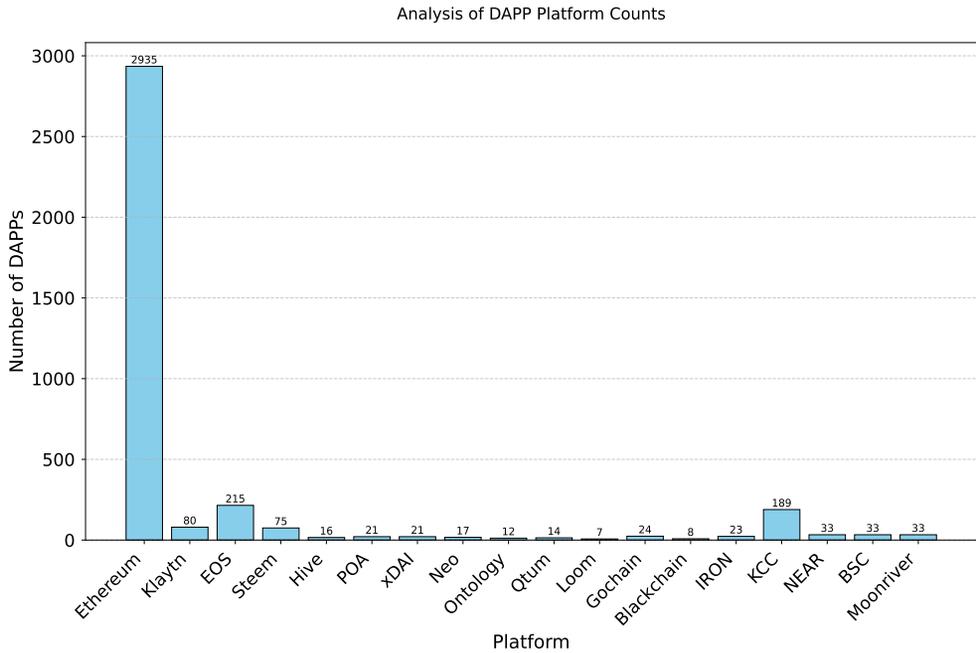


Fig. 1. Analysis on the Quantity of DApp Platforms

3.3 Category Analysis

The data in Figure.2 show that five DApp categories—gaming, gambling, social, finance, and exchanges—account for 59.4% of the total. Additionally, transaction volumes and active users, as indicated in Table 3.3.3 for Ethereum activity, are predominantly concentrated in these five categories.

From the above analysis, we can draw the following conclusions:

- Finance-related DApp (finance and exchanges) are the most popular among users and have the highest number of active users, followed by entertainment-related DApp (gaming and gambling).
- In emerging fields such as health, insurance, and energy, the number of DApp, smart contracts, and active users is relatively limited.

3.4 Reentrancy Vulnerability Detection for Smart Contracts

In the DApp development and application environment, smart contracts, which handle various business logic, are receiving increasing attention for their security issues[19, 21, 48]. Notably, the attack on The DAO, which exploited a reentrancy vulnerability in a smart contract, caused a temporary downturn in blockchain applications[30]. This highlights reentrancy attacks as a significant security threat to smart contracts.

Smart contract security is crucial for ensuring the safety and privacy of decentralized applications based on blockchain technology. Building on our analysis of DApp data behavior in the previous chapter, this chapter delves into reentrancy vulnerability attacks in smart contracts. We conduct an in-depth study of their data behavior to achieve detection and identification of reentrancy attacks.

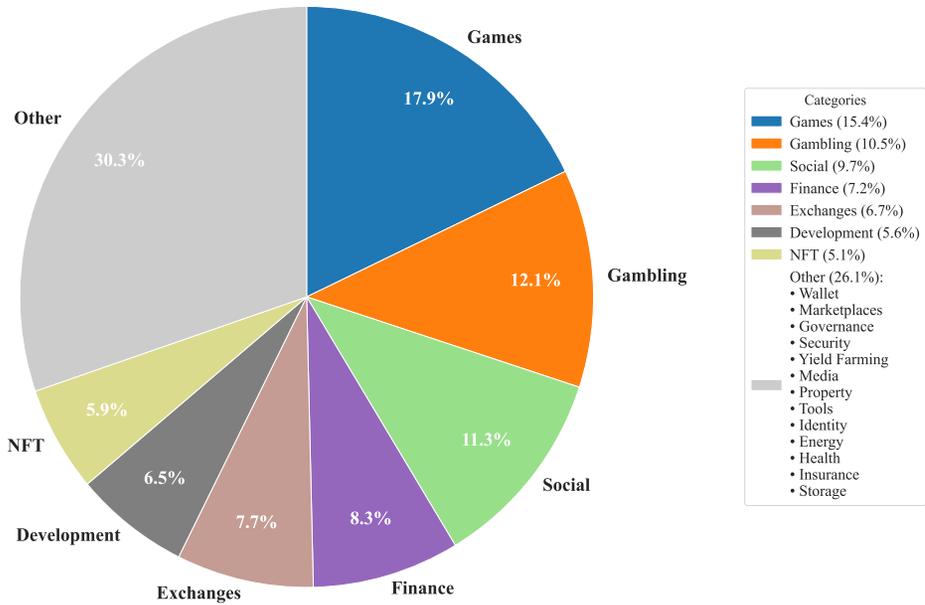


Fig. 2. DApp Category Distribution

Interest in smart contract security is growing, with researchers working to identify vulnerabilities. Traditional analysis has relied on formal methods [41, 50]. Meanwhile, advances in deep learning have expanded the role of neural networks. The LSTM model, effective for sequence tasks like speech recognition[16] and text prediction[42], is particularly noteworthy.

The method used in this chapter can secure DApp deployed on blockchain systems. The reasons are as follows:

- Smart contracts are integral to blockchain and DApp, so researching their security is vital for understanding the security of blockchain and DApp.
- Reentrancy attacks on smart contracts can be detected and identified using deep learning methods. This allows for the detection of smart contracts with reentrancy vulnerabilities, thereby mitigating the security risks faced by DApp.

We focus on the most common and severe vulnerability in EVM-based smart contracts: reentrancy attacks. This vulnerability is exploited when a contract attempts to send Ether before updating its internal state. Specifically, reentrancy attacks can occur when a function creates an external call to an untrusted smart contract.

When an attacker transfers Ether to a smart contract address, it triggers the attack contract’s fallback function[14]. Malicious code hidden in this function can activate reentrancy, causing repeated transfer operations.

In smart contracts, the fallback function is automatically triggered in two scenarios: 1) when a contract call is made but no matching function is found, it is called by default; 2) when the contract receives an Ether transfer, the fallback function can also be executed.

In the example shown in Figure.3, the attack exploits the second trigger condition of the fallback function in a smart contract, as described earlier. The money function in the attack contract attempts to execute a withdrawal by calling the withdraw function of the victim contract. This action irreversibly activates the fallback function in the attack contract, which then repeatedly executes the withdraw function until the Ether in the victim contract is depleted.

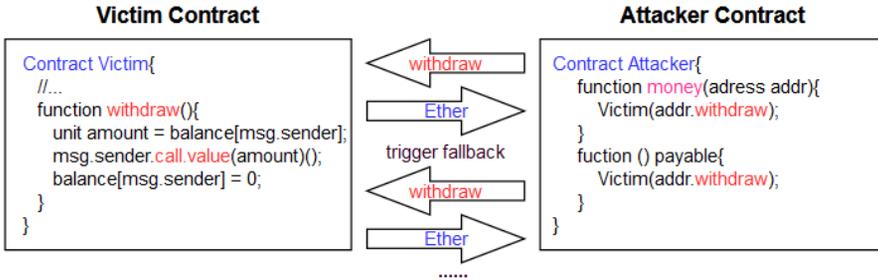


Fig. 3. Reentrancy Attack Example

3.4.1 Data Preprocessing.

Historically, there haven't been enough smart contracts to train neural networks. Today, with the increasing number of smart contracts across blockchain platforms, the time is right for neural network-based vulnerability detection. Deep learning methods are now employed to identify vulnerabilities in smart contracts.

Our objective is to automatically determine if a given smart contract is reentrant using vulnerability detection methods. The automatic reentrancy vulnerability detection process involves several steps, as shown in Figure.4. First, data cleaning of the original smart contract is essential, such as removing blank lines, non-ASCII characters, and irrelevant comments. Then, the original smart contract is converted into contract snippets composed of key program statements. Next, each contract snippet is tokenized. Each snippet is then parsed into a series of code tokens, which are embedded into feature vectors for representation. Finally, during the experimental phase, these feature vectors are input into the adopted sequential model to train the detector, thereby achieving the detection of reentrancy vulnerability attacks.

Smart contracts on Ethereum are programs written in Solidity. They consist of multiple code lines, but some lines, like comments or unrelated functions, are irrelevant for reentrancy vulnerability analysis. To facilitate precise feature extraction, we condense smart contracts into expressive contract snippets.

Since deep neural networks typically use vectors as inputs, we need to represent smart contract snippets as vectors that are semantically meaningful for reentrancy detection. First, before generating vectors for each snippet, we obtain a symbolic representation through the following steps: 1) mapping user-defined variables to symbolic names (e.g., "VAR1," "VAR2"); 2) mapping user-defined functions to symbolic names (e.g., "FUN1," "FUN2"). After this, we perform a lexical analysis to split the symbolic representation of the contract snippet into a sequence of tokens.

Then, word2vec is used to convert these tokens into vectors. Word2vec maps tokens to integers and transforms them into fixed-dimension vectors. Since contract snippets may have varying numbers of tokens, the corresponding vectors can have different lengths. To ensure uniform vector length for input, vectors are padded with zeros at the end if shorter than the fixed dimension or truncated at the end if longer.

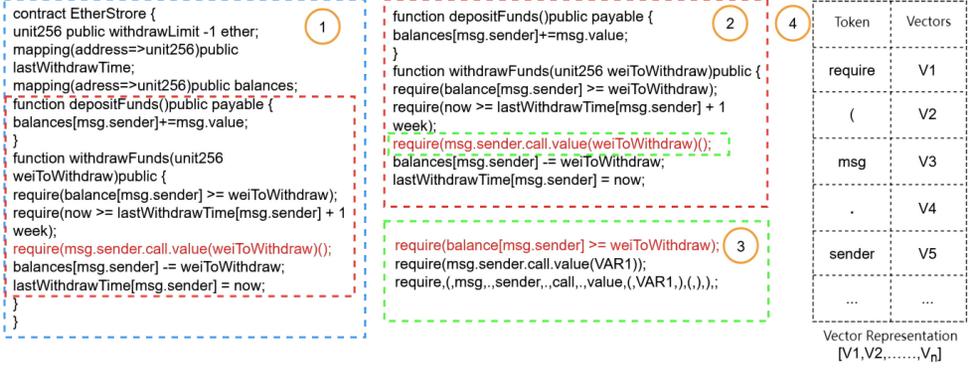


Fig. 4. Data Processing Pipeline

3.4.2 Model.

An LSTM unit consists of an input gate i_t , an output gate o_t , a forget gate f_t , and a cell state C_t , allowing the unit to remember values at any time and control information flow. As shown in Figure.5.

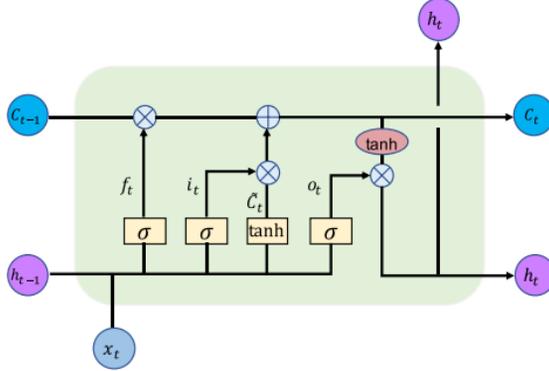


Fig. 5. LSTM Unit

The hidden state h_t of an LSTM unit can be computed as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3)$$

$$\hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (4)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \hat{C}_t \quad (5)$$

$$h_t = o_t \odot \tanh(C_t) \quad (6)$$

In this context, C_t represents a new candidate vector. The *Sigmoid* function σ and the hyperbolic tangent function \tanh are activation functions used within the LSTM unit. The symbol \odot denotes matrix multiplication and element-wise multiplication. These functions share similar equations but

differ in their parameter matrices W . Since standard LSTM cannot capture future information in a sequence, a bidirectional LSTM layer is added to address this limitation.

To highlight the importance of certain output results for vulnerability detection, we introduced an Attention Mechanism, resulting in the BLSTM-ATT sequential model. For instance, for important words in lines of code (e.g., call.value), we use the Attention Mechanism to assign weights, which can be formalized as:

$$\mu_t = \tanh(Wh_t + b) \quad (6)$$

$$\alpha_t = \frac{\exp(\mu^t \mu)}{\Sigma(\exp(\mu_t^T \mu))} \quad (7)$$

α represents a normalized weight obtained through the Attention Mechanism. The specific model architecture is shown in Figure.6.

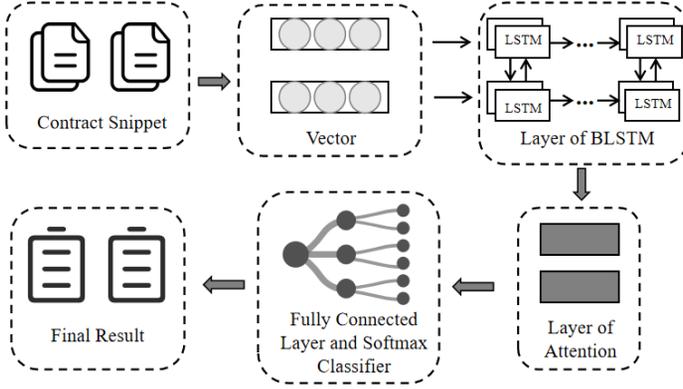


Fig. 6. Model Framework

We input word2vec-generated feature vectors into the BLSTM-ATT sequential model to learn model parameters. This involves calculating gradients and updating parameters during backpropagation. Once training is complete, the trained model is used for reentrancy detection.

Given one or more smart contract snippets from the test set, we convert them into vector representations and input these vectors into the sequential model. The model outputs results for each target smart contract, indicating whether it has reentrancy with "1" or "0".

Formally, we use a *softmax* classifier to predict the label y^* of the contract snippet S . The detector takes the hidden state h_* as input:

$$p(y/S) = \text{softmax}(Wh^* + b) \quad (8)$$

$$y^* = \text{argmax} \cdot p(y/S) \quad (9)$$

3.4.3 Model Training Details. In parameter settings, we use 10-fold cross-validation[7] to select and train the optimal parameter values for reentrancy detection. We learn the model by optimizing binary cross-entropy loss. All experiments adopt the optimal gradient descent algorithm *Adam*[13]. Our model searches for the learning rate l_r in [0.0001, 0.0005, 0.001, 0.002, 0.005]. To prevent overfitting, we adjust the dropout rate d_r searched in [0.2, 0.4, 0.6, 0.8]. The final parameters are set to: $l_r = 0.02$, $d_r = 0.2$, batch size $\beta = 64$, and vector dimension $vm = 300$.

3.4.4 *Evaluation Metrics and Experimental Results.* To evaluate the model’s performance, we use metrics such as Accuracy (ACC), True Positive Rate (TPR), False Positive Rate (FPR), Precision (PRE), and F1-score. These metrics are calculated based on four scenarios: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), as shown in Table.4.

Table 4. Table 4.5 Sample Metrics

Actual Condition	Predicted Condition	
	Positive	Negative
Positive	TP (True Positive)	FN (False Negative)
Negative	FP (False Positive)	TN (True Negative)

The formulas for the evaluation metrics are as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

$$PRE = \frac{TP}{TP + FP}$$

$$F1 = \frac{2 * PRE * TPR}{PRE + TPR}$$

For the BLSTM-ATT sequential model, we repeated the experiments 10 times to calculate the average performance, achieving favorable results. The BLSTM-ATT model achieved an F1-score of 88.26% and an FPR of 8.57%, indicating its ability to accurately identify reentrancy vulnerabilities. The high accuracy is likely due to the effectiveness of the BLSTM architecture and the attention mechanism. The BLSTM-ATT model not only captures long-term dependencies from both past and future contexts but also highlights key points through the attention mechanism.

The performance of our sequential model was further analyzed using an ROC curve[6], as shown in Figure 7. The ROC curve plots TPR on the y-axis and FPR on the x-axis and is commonly used to evaluate binary classifiers. The AUC for the BLSTM-ATT model is close to 90%, indicating good detection performance.

From the results, we can conclude that deep learning-based detection methods, specifically sequential models, achieve the detection function effectively. This indicates that deep learning can be applied to vulnerability detection in smart contracts. Additionally, due to the semantic information capture of sequential models and the highlights of the attention mechanism, vulnerability detection in smart contracts can achieve high accuracy.

The model’s good performance may be attributed to the contract snippets used, which discard useless information (such as code comments and blank lines) and capture key points (such as control flow dependencies, keywords, and semantic inheritance information). Through highly expressive contract snippets, our sequential model is well-adapted and trained to accurately identify vulnerabilities.

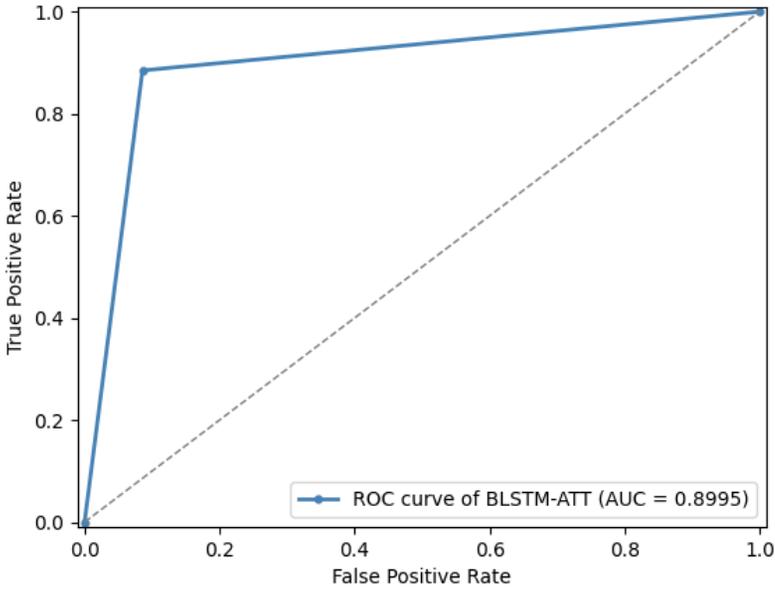


Fig. 7. ROC Curve

3.5 Privacy Protection

With the growth of cloud computing, big data, and the Internet of Things, and the digital transformation of traditional industries, data volumes are growing exponentially, marking the arrival of the big data era. As a strategic resource, data plays an increasingly important role in national governance, economic growth, and national security, so research on data privacy and security is gaining more attention. Blockchain technology, known for its decentralization, traceability, and immutability, is widely used. However, these features come at the cost of disclosing certain information, such as transaction content being exposed due to data verifiability. To reduce privacy leakage risks, privacy protection on blockchains is essential, focusing mainly on identity privacy protection and transaction privacy protection[20].

We extract privacy requirements and threats from the network environment, transactions, and applications and conduct two types of analyses: The first analysis is based on the fundamental characteristics of blockchain, while the second provides a detailed description of various threats.

3.5.1 Privacy Requirements of Blockchain.

In light of the characteristics of blockchain, we have conducted the following analyses:

- In blockchain transactions, each block contains the hash value of the previous block, forming a chained structure. This means all transactions are traceable. Therefore, we need to make the connections between transactions invisible.
- Since all transaction information is stored in a public, global ledger, any participant can access and verify all data via public interfaces. Therefore, it is essential to protect identity information, i.e., the relationship between blockchain addresses and user identity information.

The threats to blockchain privacy protection primarily stem from the associations between user identity information and blockchain addresses, as well as the transaction records and the knowledge behind them stored in the blockchain.

De-anonymization: Malicious nodes can join the network without authorization and monitor communication data at the network layer. Attackers may link transaction information captured at the network layer with the originating node’s IP address, thereby threatening user identity privacy.

- **Network Analysis:** By monitoring data transmitted in a P2P network, attackers can obtain IP addresses when nodes broadcast transactions.
- **Address Clustering:** Users can divide the network into different address clusters. After labeling with data collection techniques, some addresses can be linked to the same user. While not easy to implement, these methods remain a potential threat and cannot be ignored.
- **Denial of Service Attack:** Malicious attackers exploit insufficiencies in network security measures, rendering normal service means unusable and causing machines or network resources to become unavailable.
- **Sybil Attack:** Malicious attackers use a small number of nodes to control multiple fake identities, thereby disrupting the balance of reputation systems in a P2P network.

Transaction Pattern Analysis: Other transaction flows to the public network can be analyzed statistically. For instance, transaction graph analysis can reveal overall transaction characteristics. AS-level deployment analysis involves recursively connecting to clients, requesting, and collecting the IP addresses of other peers to gather network information. This provides specific details about the scale, structure, and distribution of the core network.

3.5.2 Identity Privacy Protection Methods.

Three common mechanisms for preserving anonymity in blockchain are: Coin Mixing[40], Ring Signatures[39], and Non-interactive Zero-Knowledge Proofs[8].

Coin Mixing: Blockchain’s transparency links transaction senders and receivers. Analyzing public data can reveal private info. A solution is to obscure transaction relationships using mixers, enhancing anonymity.

If an entity wants to send a message M to another entity at address R , they encrypt M with the recipient’s public key K_r , attach address R , and then encrypt the result with the intermediary’s public key K_1 . The left side of the following expression shows the ciphertext transferred to the intermediary:

$$K_1(r_0, K_R(r_1, M), R) \rightarrow K_R(r_1, M), R \quad (10)$$

\rightarrow represents transforming the initial ciphertext into a new ciphertext on the right. During this process, the intermediary decrypts the original ciphertext with their private key, then passes the sub-ciphertext to R , who decrypts it with their private key. Note that r_1 and r_0 , as random numbers, ensure the message isn’t transmitted multiple times.

The core idea of coin mixing is to resist transaction graph analysis without encrypting transaction content, thereby increasing the difficulty of attacks. By using intermediaries or spontaneous obfuscation to mix and transfer funds, attackers cannot directly obtain the sender-receiver correspondence in transactions, thus enhancing blockchain privacy protection.

Coin mixing techniques are categorized into centralized and decentralized approaches. Centralized methods use third-party nodes to obscure the link between transaction parties, making cryptocurrency flows harder to track. Examples include online wallets, dedicated mixing services, and multiple mixer overlays, which require no extra technical changes. Decentralized methods replace third-party nodes with a P2P protocol, combining multiple transactions into one with many inputs and outputs to hide associations.

Ring Signature: In a ring signature scheme, user A selects a group of participants, including themselves, to form a ring $user_0, user_1, \dots, user_n$. Each participant has a public key from a standard signature scheme (e.g., RSA, ECDSA). User A signs a message using their private key SK_A and all the public keys PK_0, PK_1, \dots, PK_n of the ring members. The verifier can confirm that the message was signed by one member of the group but cannot identify the actual signer. This provides complete anonymity for the signer. Ring signatures hide the signer’s identity among the public keys of the ring, with no centralized authority or administrator involved.

Non-interactive Zero-Knowledge Proof(NIZK): A zero-knowledge proof (ZKP) is a cryptographic method that allows one to prove a statement without revealing any additional information. NIZK differs from ZKP in that it requires no interaction between the prover and verifier, making it suitable for anonymous and distributed message verification in blockchain systems. A formal definition of a Non-Interactive Zero-Knowledge (NIZK) proof system is as follows: Let (P, V) denote a pair of probabilistic polynomial-time algorithms acting as the prover and verifier, respectively. For a language $\mathcal{L} \subseteq \text{NP}$ (with a security parameter k), the tuple (P, V) is called an NIZK proof system for \mathcal{L} if it satisfies the following properties:

- **Integrity:** For any input $x \in \mathcal{L}$, its witness w , and polynomial $p()$, the following must be satisfied:

$$P_r[V(R, x, P(R, x, w)) = 1] \geq 1 - \frac{1}{p(|x|)} \quad (11)$$

- **Soundness:** For any input $x \notin \mathcal{L}$, any probabilistic polynomial-time algorithm p^* , and any polynomial $P()$, the following must be satisfied:

$$P_r[V(R, x, P^*(R, x)) = 1] < \frac{1}{P(|x|)} \quad (12)$$

- **Zero-Knowledge:** For any $x \in \mathcal{L}$, its witness w , there exists a probabilistic polynomial-time simulator S such that the following distributions are computationally indistinguishable:

$$\{R, x, P(R, x, w)\} \approx \{R, x, \pi\} \leftarrow S(x) \quad (13)$$

This means that all information obtained by the verifier during interaction with the prover can also be computed by a probabilistic polynomial-time simulator. Note that R is a public random reference string.

3.5.3 Transaction Privacy Protection Methods.

Homomorphic encryption systems (HC) allow computations on ciphertexts without decrypting them. This means that operations on encrypted data yield results that match those of the same operations on plaintext. This enables tasks like querying encrypted data without exposing it, thus enhancing privacy when data is outsourced or stored with third parties.

Consider a scenario where Party A holds values (x_1, x_2, \dots, x_n) , and Party B holds a function $f()$. Both want to compute $f(x_1, x_2, \dots, x_n)$ without disclosing the values or the function’s details. In a homomorphic encryption system, A encrypts the inputs $\{E(x_1), \dots, E(x_n)\}$ and sends them to B . B performs the computation on the ciphertexts, randomizes the result, and sends it back to A . Upon decryption, A securely obtains $f(x_1, x_2, \dots, x_n)$.

Homomorphic encryption systems, as black-box operations, take n ciphertexts and operations as input and output the encrypted result of the corresponding operation on the original data. This feature makes them ideal for securely updating transaction amounts and other data in blockchains. Typical homomorphic encryption schemes for blockchain privacy protection include the Pedersen Commitment Scheme[33] and the Paillier System [31].

4 FUTURE RESEARCH DIRECTIONS

From the above chapters, we understand common privacy protection methods. This section outlines future research directions in this field.

- **Scalability:** Coin mixing causes extra waiting delays, and complex cryptographic primitives often lead to significant computational and communication overhead. These high costs limit the scalability of anonymity. Thus, one possible direction is to solve the combinatorial optimization problem between existing or new cryptographic primitives and their potential configurations.
- **Enhancing Privacy under Weaker Assumptions:** Strengthen privacy protection in scenarios with minimal or no trust assumptions.
- **Compatibility:** A major challenge is ensuring compatibility between privacy protection methods and account architectures, such as Ethereum’s account system, which maintains addresses as a global state. Ethereum, being the most widely used platform due to its built-in Turing-complete programming language, is considered ideal for DAPP development. However, integrating privacy protection with its account architecture remains a significant challenge.
- **Privacy Protection and Regulatable Control:** Blockchain’s decentralized and trustless nature has driven its widespread adoption, with privacy protection safeguarding user data. However, these technologies can be exploited for illegal activities like money laundering. Thus, blockchain activities need regulation by a trusted institution to prevent misuse while still protecting users’ sensitive data.

5 CONCLUSION

This paper focuses on the privacy and security issues of blockchain-based decentralized applications (DApp). Through data behavior analysis of DApp, we reveal their development status and conduct security analyses, supporting subsequent chapters. We also detect reentrancy vulnerability attacks in smart contracts using the BLSTM-ATT model, enabling source code-level attack detection. Furthermore, we examine privacy threats in blockchain and discuss cryptographic defenses like identity and transaction privacy protection. Finally, we summarize existing privacy protection methods and outline future research challenges in this field. Research on the privacy and security of blockchain-based decentralized applications is a vast topic. This paper has only scratched the surface by analyzing and discussing some basic knowledge, security threats, and corresponding methods. For the underlying blockchain technology of DApp, its security issues are complex and directly impact DApp security. Challenges such as data security at the data layer and consensus algorithm security at the consensus layer warrant further exploration. Additionally, smart contracts, a unique component of DApp, face not only reentrancy attacks but also other threats like short-address attacks and code injection, which will remain research hotspots.

REFERENCES

- [1] Abdikhakimov, I. (2024). The interplay of quantum computing, blockchain systems, and privacy laws: Challenges and opportunities. *Elita. uz-Elektron Ilmiy Jurnal*, 2(1):1–12.
- [2] Agarwal, U., Rishiwal, V., Tanwar, S., and Yadav, M. (2024). Blockchain and crypto forensics: Investigating crypto frauds. *International Journal of Network Management*, 34(2):e2255.
- [3] Bu, J., Li, W., Li, Z., Zhang, Z., and Li, X. (2025a). Enhancing smart contract vulnerability detection in dapps leveraging fine-tuned llm. *arXiv preprint arXiv:2504.05006*.
- [4] Bu, J., Li, W., Li, Z., Zhang, Z., and Li, X. (2025b). Smartbugbert: Bert-enhanced vulnerability detection for smart contract bytecode. *arXiv preprint arXiv:2504.05002*.
- [5] Dhillon, V., Metcalf, D., and Hooper, M. (2017). Blockchain enabled applications. *Berkeley, CA: Apress*.

- [6] Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.
- [7] Fushiki, T. (2011). Estimation of prediction error by using k-fold cross-validation. *Statistics and Computing*, 21:137–146.
- [8] Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178.
- [9] Heo, J. W., Ramachandran, G. S., Dorri, A., and Jurdak, R. (2024). Blockchain data storage optimisations: a comprehensive survey. *ACM Computing Surveys*, 56(7):1–27.
- [10] Izaguirre Diaz, I. (2024). Decentralized application (dapp) on the vechainthor blockchain for business processes. Master’s thesis, University of South-Eastern Norway.
- [11] Jimmy, F. (2024). Enhancing data security in financial institutions with blockchain technology. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, 5(1):424–437.
- [12] John, K., Monnot, B., Mueller, P., Saleh, F., and Schwarz-Schilling, C. (2025). Economics of ethereum. *Journal of Corporate Finance*, 91:102718.
- [13] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [14] Kong, D., Li, X., and Li, W. (2024). Characterizing the solana nft ecosystem. In *Companion Proceedings of the ACM Web Conference 2024*, pages 766–769.
- [15] Kovalchuk, O., Shevchuk, R., and Banakh, S. (2024). Cryptocurrency crime risks modeling: Environment, e-commerce, and cybersecurity issue. *IEEE Access*.
- [16] Kulkarni, S. P. et al. (2021). Integration of audio video speech recognition using lstm and feed forward convolutional neural network.
- [17] Li, W., Li, X., Li, Z., and Zhang, Y. (2024a). Cobra: interaction-aware bytecode-level vulnerability detector for smart contracts. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pages 1358–1369.
- [18] Li, W., Li, X., Zhang, Y., and Li, Z. (2024b). Defitail: Defi protocol inspection through cross-contract execution analysis. In *Companion Proceedings of the ACM Web Conference 2024*, pages 786–789.
- [19] Li, W., Liu, Z., Li, X., and Nie, S. (2024c). Detecting malicious accounts in web3 through transaction graph. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pages 2482–2483.
- [20] Li, X. et al. (2021). Hybrid analysis of smart contracts and malicious behaviors in ethereum. *Hong Kong Polytechnic University*.
- [21] Li, X., Yu, L., and Luo, X. (2017). On discovering vulnerabilities in android applications. In *Mobile Security and Privacy*, pages 155–166. Elsevier.
- [22] Li, Z., Li, W., Li, X., and Zhang, Y. (2024d). Guardians of the ledger: Protecting decentralized exchanges from state derailment defects. *IEEE Transactions on Reliability*.
- [23] Li, Z., Li, W., Li, X., and Zhang, Y. (2024e). Stateguard: Detecting state derailment defects in decentralized exchange smart contract. In *Companion Proceedings of the ACM Web Conference 2024*, pages 810–813.
- [24] Li, Z., Li, X., Li, W., and Wang, X. (2025). Scalml: Detecting bad practices in smart contracts through llms. *arXiv preprint arXiv:2502.04347*.
- [25] Liu, Z., Huang, B., Li, Y., Sun, Q., Pedersen, T. B., and Gao, D. W. (2024a). Pricing game and blockchain for electricity data trading in low-carbon smart energy systems. *IEEE Transactions on Industrial Informatics*, 20(4):6446–6456.
- [26] Liu, Z. and Li, X. (2025). Sok: Security analysis of blockchain-based cryptocurrency. *arXiv preprint arXiv:2503.22156*.
- [27] Liu, Z., Li, X., Peng, H., and Li, W. (2024b). Gastrace: Detecting sandwich attack malicious accounts in ethereum. In *2024 IEEE International Conference on Web Services (ICWS)*, pages 1409–1411. IEEE.
- [28] Mao, Y., Li, X., Li, W., Wang, X., and Xie, L. (2024). Scla: Automated smart contract summarization via llms and control flow prompt. *arXiv preprint arXiv:2402.04863*.
- [29] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.
- [30] Niu, Y., Li, X., Peng, H., and Li, W. (2024). Unveiling wash trading in popular nft markets. In *Companion Proceedings of the ACM Web Conference 2024*, pages 730–733.
- [31] Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*, pages 223–238. Springer.
- [32] Patel, N. A., Ingle, P. S., Patsamatla, S. K., Omotunde, H., Ingole, B. S., et al. (2024). Integration of blockchain and ai for enhancing data security in healthcare: A systematic review. *Library of Progress-Library Science, Information Technology & Computer*, 44(3).
- [33] Pedersen, T. P. (1991). Non-interactive and information-theoretic secure verifiable secret sharing. In *Annual international cryptology conference*, pages 129–140. Springer.
- [34] Popoola, O., Rodrigues, M., Marchang, J., Shenfield, A., Ikpehai, A., and Popoola, J. (2024). A critical literature review of security and privacy in smart home healthcare schemes adopting iot & blockchain: problems, challenges and solutions. *Blockchain: Research and Applications*, 5(2):100178.

- [35] Priya, J. C., Praveen, R., Nivitha, K., and Sudhakar, T. (2024). Improved blockchain-based user authentication protocol with ring signature for internet of medical things. *Peer-to-Peer Networking and Applications*, 17(4):2415–2434.
- [36] Radanliev, P. (2024). The rise and fall of cryptocurrencies: defining the economic and social values of blockchain technologies, assessing the opportunities, and defining the financial and cybersecurity risks of the metaverse. *Financial Innovation*, 10(1):1.
- [37] Rao, I. S., Kiah, M. M., Hameed, M. M., and Memon, Z. A. (2024). Scalability of blockchain: a comprehensive review and future research direction. *Cluster Computing*, 27(5):5547–5570.
- [38] Ray, R. K., Chowdhury, F. R., and Hasan, M. R. (2024). Blockchain applications in retail cybersecurity: Enhancing supply chain integrity, secure transactions, and data protection. *Journal of Business and Management Studies*, 6(1):206–214.
- [39] Rivest, R. L., Shamir, A., and Tauman, Y. (2001). How to leak a secret. In *Advances in Cryptology—ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings 7*, pages 552–565. Springer.
- [40] Ruffing, T., Moreno-Sanchez, P., and Kate, A. (2014). Coinshuffle: Practical decentralized coin mixing for bitcoin. In *Computer Security-ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7–11, 2014. Proceedings, Part II 19*, pages 345–364. Springer.
- [41] Samreen, N. F. and Alalfi, M. H. (2020). Reentrancy vulnerability identification in ethereum smart contracts. In *IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, pages 22–29. IEEE.
- [42] Santhanam, S. (2020). Context based text-generation using lstm networks. *arXiv preprint arXiv:2005.00048*.
- [43] Shen, Z., Wang, Y., Wang, H., Liu, P., Liu, K., and Liu, M. (2024). Privacy-protecting predictive cache method based on blockchain and machine learning in internet of vehicles. *Vehicular Communications*, 47:100771.
- [44] Singh, S., Hosen, A. S., and Yoon, B. (2021). Blockchain security attacks, challenges, and solutions for the future distributed iot network. *Ieee Access*, 9:13938–13959.
- [45] Taylor, P. J., Dargahi, T., Dehghantanha, A., Parizi, R. M., and Choo, K.-K. R. (2020). A systematic literature review of blockchain cyber security. *Digital Communications and Networks*, 6(2):147–156.
- [46] Tolmach, P., Li, Y., Lin, S.-W., Liu, Y., and Li, Z. (2021). A survey of smart contract formal specification and verification. *ACM Computing Surveys (CSUR)*, 54(7):1–38.
- [47] Torres, C. F., Iannillo, A. K., Gervais, A., and State, R. (2021). Confuzzius: A data dependency-aware hybrid fuzzer for smart contracts. In *IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 103–119. IEEE.
- [48] Wang, Y., Li, X., Ye, S., Xie, L., and Xing, J. (2024). Smart contracts in the real world: A statistical exploration of external data dependencies. *arXiv preprint arXiv:2406.13253*.
- [49] Wu, G., Wang, H., Lai, X., Wang, M., He, D., and Chan, S. (2024). A comprehensive survey of smart contract security: State of the art and research directions. *Journal of Network and Computer Applications*, page 103882.
- [50] Xue, Y., Ma, M., Lin, Y., Sui, Y., Ye, J., and Peng, T. (2020). Cross-contract static analysis for detecting practical reentrancy vulnerabilities in smart contracts. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*, pages 1029–1040.
- [51] Zhang, S., Zheng, K., and Wang, B. (2024). A v2v electricity transaction scheme with privacy protection based on the internet of vehicles and consortium blockchain. *International Journal of Electrical Power & Energy Systems*, 157:109789.