Indonesian Journal of Electrical Engineering and Computer Science
Vol. 40, No. 2, November 2025, pp. 1023~1039
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v40.i2.pp1023-1039

A deep learning-integrated proxy model for efficient cryptocurrency payments

Vinay Kumar Kasula, Akhila Reddy Yadulla, Bhargavi Konda, Mounica Yenugula, Supraja Ayyamgari
Department of Information Technology, University of the Cumberlands, Kentucky, USA

Article Info

Article history:

Received Nov 13, 2024
Revised Aug 2, 2025
Accepted Oct 15, 2025

Keywords:

Blind signature algorithm
Blockchain technology
Decentralized cryptocurrency
Fraud detection
Privacy protection
Proxy-based payment system
deep learning

ABSTRACT

Blockchain technology allows decentralized cryptocurrencies to change digital finances by providing secure, pseudonymous transactions to users. Since blockchain ledgers operate in a public environment, users can face potential privacy risks due to the exposure of their transaction patterns. Conventional cryptocurrency systems use block generation for transaction confirmation, yet this process produces latency and impacts the real-time efficiency of transactions. This paper develops a proxy-assisted cryptocurrency payment system that employs blind signature principles to achieve better system privacy and enhanced speed. The core functionality of this proposed system aims to protect transaction secrecy as it speeds up confirmation processes. A proxy node handles transaction requests through blind signature protocols that guarantee data confidentiality as part of the methodology. The proposed system utilizes deep learning tools, which include recurrent neural networks (RNN), graph neural networks (GNN), and reinforcement learning (RL) to forecast confirmation results, identify scams, and control proxy functions dynamically. Research indicates that the introduced method substantially boosts privacy features, decreases transaction latencies, and enhances the security of all transactions by providing an encouraging roadmap for secure cryptocurrency systems that preserve privacy.

Corresponding Author:

Vinay Kumar Kasula
Department of Information Technology, University of the Cumberlands
6178 College Station Dr, Williamsburg, Kentucky, USA
Email: vkasula19501@ucumberlands.edu

1. INTRODUCTION

Cryptocurrency, a product of the rapid advancement of modern computer communication technology, represents the evolution of money as a means of payment. Cryptocurrencies leverage cryptographic techniques to ensure the security and efficiency of digital currencies, greatly enhancing convenience in people's daily lives. Among the various forms of cryptocurrencies, those based on blockchain technology have become the most popular. The implementation of lightweight simplified payment verification (SPV) clients [1], [2] makes these decentralized cryptocurrencies even more practical. Decentralized cryptocurrencies were proposed in 2015 [3] and saw their first transaction in May 2014 [2]. The decentralized nature of these cryptocurrencies means that their issuance and transactions do not rely on any central financial authority but instead follow a peer-to-peer model. They use a public ledger, or blockchain, to record transactions, which prevents double-spending, while currency issuance is managed through network nodes' computations. This approach provides cryptocurrencies with good monetary performance, similar to gold, and fundamentally addresses inflation issues. The blockchain is maintained by

anonymous participants, known as miners, who sustain and extend the blockchain by executing a consensus protocol. The protocol's execution involves creating new blocks, a process that requires miners to solve complex mathematical problems known as proof of work (PoW). Each time a miner generates a new block, they receive a reward, which results in the creation of new currency. The blockchain system automatically adjusts the difficulty of these mathematical problems to ensure that a new block is generated approximately every 10 minutes, containing legitimate transaction information verified by the miners. Once a transaction is published on the blockchain and k additional blocks are generated thereafter, it is considered valid [4], effectively preventing an attacker from engaging in double-spending [5]. Specifically, if an attacker attempts to alter a transaction record, they must recalculate the solution to the mathematical problem for the affected block. Furthermore, any change in the hash value of this block would necessitate recalculating the solutions for all subsequent blocks. To have the tampered chain accepted by most miners, the attacker must generate a chain at least as long as the legitimate chain, which would require at least 51% of the network's computational power [6]. As k increases, the probability of a successful double-spend attack decreases exponentially. Generally, a transaction is considered valid when k = 6 [3].

While secure, blockchain technology introduces inherent transaction delays due to its structural properties. To validate a payment securely, at least six blocks must be generated, resulting in a minimum confirmation time of about one hour, as each block requires approximately 10 minutes to be created. This delay is significant for users seeking faster transactions. Additionally, using blockchain as a public ledger, while beneficial for transparency, raises privacy concerns. Although users operate under pseudonyms, with public key accounts not directly tied to personal identities, this pseudonymity is limited. An attacker could trace users' IP addresses or analyze transaction patterns and network topology on the blockchain to infer identities, potentially compromising user privacy (as noted in sources [7]-[9]). As blockchain networks grow, so does the amount of publicly accessible transaction data, making de-anonymization and privacy risks even greater. To address these challenges, this paper proposes introducing a proxy as an intermediary for cryptocurrency payments, a solution that not only mitigates transaction delays but also strengthens privacy. The proxy acts as a trusted intermediary with its own public key address, significantly reducing transaction confirmation times by facilitating faster exchanges. Furthermore, privacy is reinforced through the use of partially blind signature algorithms and one-time public key addresses, which protect user identities by making it harder to trace transactions back to individuals. This proxy-based model is designed to seamlessly integrate with existing blockchain protocols, preserving full compatibility while enhancing user privacy and efficiency. The approach leverages the proxy to handle payment processing without altering the underlying blockchain protocol, offering a practical solution that maintains the blockchain's integrity and security features.

Integrating deep learning algorithms can greatly improve the overall effectiveness of this approach by providing enhanced capabilities for detecting privacy breaches and fraudulent activities through advanced data analysis. Deep learning models can process blockchain data more efficiently, identifying complex patterns that might indicate potential security risks. Specifically, recurrent neural networks (RNNs) and graph neural networks (GNNs) are well-suited for analyzing sequential and relational data, respectively. These models can track transaction patterns, identify irregular behaviors, and even predict anomalous activities, as demonstrated by Johnson *et al.* [10]. Additionally, reinforcement learning techniques can be applied to optimize the proxy mechanism dynamically, adjusting parameters in real time to improve both transaction speed and security. This approach is supported by research from Hu *et al.* [11], which highlights the potential of reinforcement learning to make systems more adaptive and responsive. By integrating deep learning and reinforcement learning with the proposed proxy-based model, the cryptocurrency payment system could become significantly more secure, efficient, and privacy-preserving. This combination of advanced technologies

2. BACKGROUND KNOWLEDGE

2.1. Fundamental principles

In this section, we lay the groundwork for understanding the fundamental principles crucial to the topic. We begin by exploring the essential concepts, terminologies, and frameworks that underpin this area of study, establishing a basis for the methodologies and systems analyzed in the following sections. These foundational principles are central to developing and evaluating effective solutions within the field. Cryptocurrencies that rely on blockchain technology involve three primary components: transaction construction, consensus protocols, and a communication network. Each of these components plays a critical role in ensuring the security, efficiency, and scalability of cryptocurrency systems, and each leverages cryptographic algorithms and, increasingly, advanced techniques like deep learning algorithms.

Transaction construction: Transaction construction employs cryptographic algorithms such as the elliptic curve digital signature algorithm (ECDSA) and SHA-256. Each transaction comprises inputs, outputs,

and related parameters. The input is the public key address of the sender, while the output is the public key address of the recipient (which is essentially a hashed value of the public key). The amount of currency in the input must be greater than or equal to the amount in the output; the difference is considered the transaction fee, which is awarded to the miner. The signature in the transaction is created by the sender using their private key, which serves as one of the bases for verifying the legality of the transaction. For instance, Figure 1 illustrates a basic transaction structure. On the left side, transaction A represents the payment address, while B and C are the recipient addresses, with σA being the signature of A's owner. On the right side, B serves as both a payment and a recipient address, D is a recipient address, and σB represents B's owner's signature.

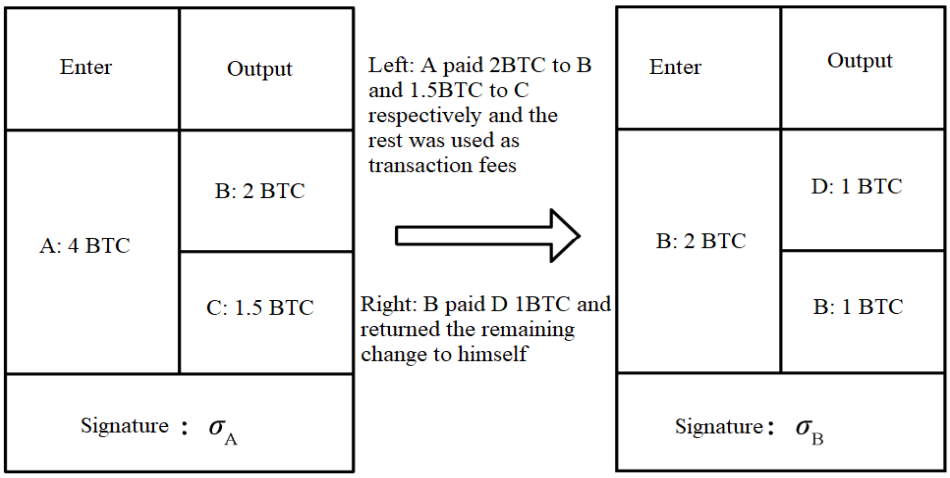

Figure 1. Transaction structure

Consensus protocol: In blockchain systems, the consensus protocol is a fundamental set of rules that miners follow to validate, maintain, and extend the blockchain. The process begins when network nodes create and broadcast transactions, which miners then collect and verify for authenticity. This verification includes checking the correctness of digital signatures, ensuring that transaction input addresses are legitimate, and confirming that the currency amounts are valid. Once validated, miners assemble these verified transactions into a data block, creating what is known as a candidate block. To secure the candidate block, miners undertake extensive computational work to solve a cryptographic puzzle, typically by calculating a specific hash function to find the correct "Nonce" value. Once a miner successfully discovers this Nonce value, the candidate block is broadcast to the entire network for validation. Other miners then examine the block's Nonce and related data to confirm its validity. If verified, the block is accepted and added to the blockchain as the latest link in the chain, becoming the "parent" block for subsequent candidate blocks. This decentralized verification and extension of the blockchain by miners ensures both the integrity and security of blockchain records.

Communication network: In blockchain systems, both transactions and blocks are broadcast through a decentralized, peer-to-peer (P2P) communication network. This network consists of equal-status nodes operating without a central authority, and it employs a randomized network topology to distribute data evenly across nodes. At the core of the network's communication strategy is a flooding algorithm, which allows each node to relay messages to its peers. This process enables all nodes to receive updates within seconds, ensuring that the blockchain remains synchronized and resilient. The efficiency and security of this communication network can be significantly enhanced by incorporating advanced deep-learning techniques. For example, reinforcement learning algorithms have shown potential in optimizing consensus protocols. By dynamically adjusting network parameters and node behaviors, reinforcement learning can help minimize latency and improve the reliability of consensus processes, as evidenced in studies such as those by [1] and [12]. Similarly, adversarial learning has proven effective in detecting and mitigating threats from malicious nodes. By identifying suspicious activity patterns or irregular network behaviors, adversarial learning algorithms help strengthen network defenses against attacks, as demonstrated in the research by [3] and [13]. Integrating these machine learning approaches into the blockchain's communication network enables nodes to not only react to potential threats but also adapt to them proactively. Consequently, this synergy between decentralized network architecture and deep learning algorithms creates a robust framework that supports

more efficient, secure, and resilient blockchain operations, ultimately enhancing the performance and trustworthiness of the entire ecosystem.

2.2. Elliptic curve digital signature algorithm (ECDSA)

The elliptic curve digital signature algorithm (ECDSA) is a cryptographic algorithm employed to ensure the integrity and authenticity of transactions in blockchain networks. ECDSA involves four crucial steps: parameter generation, key generation, signature generation, and signature verification. In the blockchain context, the ECDSA typically utilizes the secp256k1 elliptic curve, known for its efficiency and security, with specific parameters detailed in [12]. During the parameter generation phase, elliptic curve parameters are established, including the curve's base point and modulus, ensuring the cryptographic operations are robust and secure. The key generation process involves selecting a private key from a predefined set of integers, which is then used to generate the corresponding public key through elliptic curve point multiplication. In the signature generation step, a user signs a transaction by applying their private key to a cryptographic hash of the transaction data, producing a digital signature composed of two values, 'r' and 's.' Finally, during signature verification, the recipient uses the sender's public key to confirm the validity of the signature by performing elliptic curve calculations that validate the authenticity of the transaction. This process ensures that transactions are secure, verifiable, and resistant to tampering, making ECDSA a foundational element in blockchain security protocols.

The setup begins with generating system parameters, yielding public parameters (q, p, F_q, a, b, G, n), where a and b define the elliptic curve, F_q is the finite field, G the base point, and n the order of$G$.. In Key$Gen(d, pp)$, a random integer $d \in [1, n-1]$ is selected to compute$Q = dG$, where Q is the public key and d is the private key, producing a public-private key pair using the public parameters pp and integer d. $Sig(d, m)$, the signature generation algorithm uses the private key d and message m as inputs to output a signature for the message m. The Verify$(Q, Sig(d, m))$ algorithm then takes the public key Q and the signature of m as inputs, outputs 1 if the signature is valid and 0 if not.

2.3. Partial blind signature algorithm

The partial blind signature algorithm [13] consists of four main algorithms: system generation, key generation, signature issuance, and signature verification. The signature issuance algorithm, in particular, involves four steps: message blinding, consensus message generation, signing, and unblinding. It is an interactive protocol between the message owner and the signer. The steps and symbols for the signature issuance and verification algorithms are as follows:

$Blind(m)$, is the message blinding algorithm. It takes a message m to be signed as input and produce a blinded message m^* as output. This blinding process ensures that the signer cannot see the original message, preserving the privacy of the content being signed.

$\tau(c)$ refers to the consensus message generation algorithm. This consensus message generation algorithm is denoted as $\tau(c)$, plays a crucial role in ensuring that participants in a network reach an agreement on a particular transaction or data. It takes as input a parameter, c, which can represent transaction details or other consensus-related information that needs to be validated within the system. Once the algorithm processes the input, it outputs a consensus message, $\tau(c)$, which signifies that the signer has confirmed their agreement with the content and context of the message. This process is essential for maintaining consistency and trust across decentralized systems or networks. By ensuring that all participants are in agreement, the algorithm facilitates secure and reliable transactions in blockchain and distributed ledger systems.

$BldSig(sk, \tau(c), m*)$: Signature generation algorithm. This algorithm takes the private key sk, the consensus message $\tau(c)$, and the blinded message m^* as inputs. Using these inputs, it generates a partial blind signatureσ^*, which is a cryptographic signature that binds the message to the consensus while keeping the original content hidden.

$1/Blind(\sigma^*)$: Unblinding algorithm: This algorithm is an unblinding process that reverses the blinding applied to a message in the earlier stages of the protocol. It takes a partially blinded signature σ^* as its input, which was generated during the initial signing process. The algorithm then removes the blinding factor, effectively returning the original signature σ for the message m. This final signature can be used for verification purposes, ensuring the integrity and authenticity of the signed content. Despite revealing the signature, the unblinding process preserves the anonymity of the original message, maintaining privacy while allowing for secure verification.

$Rl(m, \tau(c))$: The relation function plays a critical role in ensuring the integrity of signatures within a consensus-based system. It takes two inputs: the message m and the consensus message $\tau(c)$, both of which are necessary to generate a valid signature. The function produces a signature, σ, which confirms that it is the

correct signature for the given message m in the context of the consensus message $\tau(c)$. This relationship between the message and the consensus context is crucial for validating the authenticity of the signature. Essentially, the function guarantees that the signature is not only valid for the message but also aligns with the broader consensus agreement, ensuring both integrity and trust within the system.

$BldVer(pk, (\sigma))$: Signature verification algorithm takes the public key pk and the signature σ as inputs, the algorithm checks the validity of the signature; if the signature is correct and valid, it outputs 1, confirming the authenticity of the message and signature. Otherwise, it outputs 0, signaling that the signature is invalid.

The combination of these algorithms creates a powerful system designed to facilitate blind signing, which ensures that sensitive information remains private while maintaining the integrity of the signature. By utilizing blind signing, the system allows for the authentication of transactions without exposing the underlying data, ensuring that privacy is preserved throughout the process. This method is especially beneficial in blockchain environments where anonymity is crucial, as it enables secure, verifiable transactions without compromising user confidentiality. The consensus-based framework ensures that all participants in the system agree on the validity of the signature, adding an extra layer of trust and security. Ultimately, these algorithms provide a secure way to conduct transactions, ensuring both privacy and authenticity while protecting sensitive data in decentralized applications.

2.4. Anonymity techniques

Blockchain, being a public ledger, is open and transparent, making it susceptible to privacy issues. As anyone can access the data on the blockchain, sensitive information such as transaction details and user identities could be exposed. To mitigate these privacy concerns, two main strategies have emerged to enhance anonymity: altcoin-based methods and coin-mixing techniques.

2.4.1. Altcoin-based methods

Altcoin-based methods aim to improve anonymity by converting a cryptocurrency into a substitute coin, known as an altcoin, and later converting it back to the original cryptocurrency. This method hides the identity of the user by obscuring the transaction trail through the use of alternative cryptocurrencies. The Zerocoin protocol [14] and the Zerocash protocol [15] are prominent examples of such techniques. Both protocols leverage cryptographic techniques to ensure that transactions are untraceable and unlinkable, thereby providing enhanced privacy for users. While these methods are effective at strengthening anonymity, their integration comes with significant challenges. The introduction of substitute coins modifies the underlying blockchain protocol, making them incompatible with existing systems. This incompatibility means that these privacy-enhancing protocols require substantial changes to the blockchain's infrastructure, which can limit their adoption and scalability in widely used blockchain networks.

2.4.2. Coin mixing without a central mixing authority

Coin mixing without a central authority is a decentralized method designed to enhance privacy by allowing users to mix their coins together in a way that obscures the transaction trail. In this approach, multiple participants collectively generate a coin-mixing transaction. Each participant contributes their pre-mixed coin addresses as inputs, while the outputs of the transaction correspond to the target addresses. The key feature of this method is that the transaction is only valid if it includes signatures from all involved participants, ensuring that all users are involved in the mixing process. A well-known protocol for this type of coin mixing is CoinJoin [16], which facilitates the creation of a single transaction that combines inputs from various users, thereby preventing the linkage of inputs to specific outputs and ensuring the anonymity of the participants. CoinJoin successfully provides anonymity by making it difficult for external observers to associate specific inputs with their corresponding outputs. However, despite its effectiveness, CoinJoin is vulnerable to Denial-of-Service (DoS) attacks. In such attacks, malicious users may disrupt the mixing process by either refusing to sign or participating in the mixing transaction with the intention of blocking the process. This vulnerability poses a threat to the reliability and efficiency of the protocol. To address this issue, CoinShuffle [17] was developed as an improvement to CoinJoin. CoinShuffle introduces an ordering protocol that enhances the mixing process by ensuring that the order in which participants contribute their inputs is randomized, making it harder to predict the final transaction outputs. Additionally, CoinShuffle incorporates an accountability protocol designed to mitigate DoS attacks. This protocol helps track the behavior of participants, making it possible to identify and exclude malicious actors who attempt to disrupt the mixing process. By enhancing the security and reliability of the coin mixing procedure, CoinShuffle improves upon CoinJoin, making it a more robust solution for ensuring user privacy in blockchain transactions.

2.4.3. Coin mixing with a central mixing authority

Coin mixing with a central mixing authority is an approach that centralizes the coin-mixing process, relying on a trusted intermediary to handle the mixing of coins for multiple users. This method enhances the overall reliability and scalability of the mixing process, making it more suitable for large-scale operations. By utilizing a central authority, this technique provides stronger resistance to Denial-of-Service (DoS) attacks, which can plague decentralized mixing systems, as it can ensure the continuous availability and integrity of the mixing service. One of the prominent protocols that use a central mixing authority is MixCoin [17]. MixCoin employs multiple mixing networks to perform multi-level mixing, where coins are mixed through several layers, ensuring that the central authority cannot correlate user input addresses with output addresses. This approach increases the complexity of tracing individual transactions, thus improving user anonymity. The multiple layers of mixing add an additional level of security, making it more difficult for any single party to break the anonymity of the users. Building upon MixCoin, the BlindCoin protocol [18] further strengthens privacy by introducing a public ledger that ensures the reliability and transparency of the mixing center, helping to establish trust in the central authority. BlindCoin also utilizes blind signature algorithms, which prevent the mixing center from seeing the user's output address. This addition eliminates the need for multi-level mixing, simplifying the process while maintaining a high level of security. By using blind signatures, BlindCoin ensures that even the central authority cannot track or associate specific user inputs with their corresponding outputs, further enhancing users' privacy. To improve the effectiveness and security of these coin-mixing techniques, deep learning algorithms can be integrated into the system. For instance, generative adversarial networks (GANs) [19] can be used to detect anomalies in the mixing process, identifying patterns that could suggest malicious activity or attempts to deanonymize users. Additionally, reinforcement learning [20] can be applied to optimize coin-mixing strategies, dynamically adjusting the mixing process to ensure maximum privacy and efficiency while preventing potential vulnerabilities. By incorporating these advanced techniques, the coin mixing system can provide stronger privacy protection, better performance, and increased resilience to attacks.

3. MIXING SYSTEM WITH A MIXING CENTER

The mixing system with a mixing center is based on the mixing center model in the MixCoin protocol. The primary aim of the MixCoin protocol is to enhance anonymity through mixing, while the core objective of this system is to utilize intermediaries that prevent "double spending" to improve transaction efficiency [21]. In this paper, we will focus on the core aspects of the MixCoin protocol and introduce a deep learning algorithm to further enhance efficiency while omitting unrelated details.

3.1. MixCoin system model

The MixCoin system is designed to facilitate the anonymous transfer of cryptocurrency through a coin-mixing protocol. It involves two primary entities: the mixing center (S) and the user (Alice). These two components interact to ensure the privacy and security of transactions, preventing the correlation of a user’s original and destination accounts [22]. The system functions as follows:

Mixing center (S): The mixing center, denoted as S, plays a crucial role in the MixCoin protocol. It is responsible for facilitating the coin-mixing process by collecting and mixing coins from various users. The mixing center operates with a pair of long-term signature keys, consisting of a public key and a private key. The public key is used to verify transactions, while the private key is used for signing transactions and ensuring their authenticity. The reputation of the mixing center is an important factor influencing its effectiveness and user trust. A reputable mixing center attracts more users, as it guarantees a higher level of security and privacy. Conversely, a mixing center with a poor reputation is less likely to be trusted, resulting in fewer users and potential difficulties in maintaining a secure and anonymous mixing environment. The reputation of S is often tied to its history of successfully maintaining user anonymity and preventing any leaks of transaction information.

User (Alice): Alice, the user in this system, holds a certain amount of cryptocurrency in an account. This account may be linked to her real-world identity, which presents a risk of privacy violation. Alice’s goal is to transfer the cryptocurrency to a new account while ensuring that there is no way for third parties or attackers to trace the link between the old and new accounts. In the MixCoin system, Alice sends her cryptocurrency to the mixing center, which will mix her funds with those of other users to create a pool of transactions. Once the mixing process is complete, Alice receives an equivalent amount of cryptocurrency in the form of newly generated tokens, which are sent to her new account. This process ensures that the transaction history of Alice’s original account is obfuscated, making it difficult or impossible for attackers to establish a connection between the old and new accounts.

The MixCoin system enhances privacy in cryptocurrency transactions by combining a mixing center with the user's interaction, ensuring anonymous transfers. The mixing center, which acts as a trusted central authority, plays a pivotal role in obscuring the origin and destination of cryptocurrency transactions. By using advanced cryptographic techniques such as mixing and re-routing transactions, the system ensures that users like Alice can send funds without exposing their identity or transaction details. This method of mixing funds reduces the risk of transaction tracing, making it much more difficult for third parties to track the movement of the funds across the network. Ultimately, the MixCoin system strengthens user privacy and security while preserving the efficiency and functionality of cryptocurrency transactions.

3.2. MixCoin protocol with deep learning enhancements

The MixCoin protocol, designed to enhance privacy and anonymity in cryptocurrency transactions, has been significantly improved with the integration of deep learning algorithms. These deep learning optimizations aim to boost the efficiency and reliability of the protocol by automating and optimizing key processes such as transaction mixing, which obfuscates the origin of funds. The updated protocol utilizes deep learning to predict transaction patterns, detect anomalies, and dynamically adjust the mixing process to maintain high levels of privacy [23]. Furthermore, these algorithms enable faster and more accurate transaction processing, reducing delays and increasing overall system throughput. As a result, the enhanced MixCoin protocol provides a more robust and scalable solution for users seeking enhanced anonymity in their cryptocurrency transactions. The key steps of the protocol, now integrated with deep learning optimizations, are as follows:

3.2.1. Request initiation

The protocol begins with Alice, the user seeking anonymity, sending a mixing request to the mixer, S (a service that mixes the currency). The request includes several parameters:K_{in}: Alice's source address before the mixing process, K_{out}: Alice's destination address after mixing, w: The amount of currency Alice wants to mix, t_1: The deadline by which Alice must transfer the currency to S, t_2: The deadline by which S must return the currency to Alice, K_{esc}: An escrow address controlled by S for added security, D and D': System parameters that configure the transaction settings for this session. These parameters are securely sent to S, initiating the mixing request and setting the transaction terms for both parties.

3.2.2. Request acceptance

When S receives Alice's request, it evaluates the parameters using a deep learning model designed to predict transaction trends and potential network delays. This model helps S make an informed decision, especially when evaluating factors like transaction size, timing, and current network congestion. If S decides to accept the request, it confirms this by sending Alice the same parameters along with its signature K_S as proof of acceptance. This signature confirms $S's$ commitment to processing the transaction under the agreed-upon conditions. If, however, S rejects the request (due to reasons like high network congestion or risk factors identified by the deep learning model), both parties exit the protocol without further steps. If S accepts Alice's mixing request, it uses a deep learning model to predict transaction patterns and potential delays, optimizing its response. S sends $K_{in}, K_{out}, w, t_1, t_2, K_{esc}, D, D'$ and its signature K_S back to Alice. If S rejects the request for any reason, both parties exit the protocol.

3.2.3. Payment processing

Once the request is accepted, Alice must transfer the specified amount w to S's account K_{esc} before the deadline t_1. This payment is recorded on the blockchain, creating an auditable trail of the transaction. To further optimize this step, the deep learning model assesses real-time network conditions to predict potential congestion and minimize delays in confirming Alice's payment on the blockchain. If Alice fails to make the payment within the deadline, S automatically exits the protocol. This protocol step ensures that only committed transactions are processed, enhancing security and reducing wasted resources.

3.2.4. Currency return

The final step involves S returning the equivalent amount of currency to Alice's destination address K_{out} before the deadline t_2. The transaction, once completed, is recorded on the blockchain to maintain transparency. During this process, the deep learning model optimizes the timing of S's return transaction, aiming for promptness while mitigating network-related delays. This model also improves the accuracy of compliance by analyzing prior transactions and adjusting transaction parameters to maintain efficiency. In the event of a delay or if Alice does not receive the funds on time, she can reveal the signed commitment K_S from Step 2, proving her compliance with the protocol. This mechanism allows Alice to protect her funds and reputation while holding S accountable, as a breach could harm S's credibility within the system.

The integration of deep learning in the MixCoin protocol brings significant improvements. Predicting network conditions: By forecasting congestion and delays, the model allows both parties to anticipate and manage transaction times effectively. Optimizing throughput: The model helps prioritize transactions, improving the speed and reliability of currency mixing. Enhanced efficiency: By predicting and reducing potential bottlenecks, the deep learning model enhances the protocol's overall performance, making it faster and more resilient under varying network conditions. This enhanced MixCoin protocol combines traditional cryptographic mixing techniques with advanced deep learning models to deliver a robust, efficient, and user-centric solution for cryptocurrency transactions that require high levels of privacy and anonymity.

3.3. MixCoin protocol with deep learning enhancements

The MixCoin protocol is a privacy-preserving solution aimed at enabling anonymous cryptocurrency transactions by obscuring the origin of funds through a central mixing entity referred to as the "mixing center." By adding deep learning algorithms, the protocol achieves an improved ability to manage and respond to transaction requests, providing greater efficiency, minimizing latency, and optimizing the mixing process for a smoother user experience [24]. This integration with deep learning adds predictive insights to each stage, from request initiation to fund return, thus enhancing MixCoin's core functionality. Below is an in-depth outline of the enhanced MixCoin protocol with its key operational steps:

3.3.1. Request initiation

Alice begins the process by sending a set of parameters to the mixing center (S). These parameters include her K_{in} (the account she holds before mixing), K_{out} (the account she wishes the funds to be sent to after mixing), w (the amount of currency to be mixed), t_1 (the deadline for Alice to pay the mixing center), t_2 (the deadline for S to return the funds), K_{esc} (S's receiving address), D and D' (system parameters). This initial request triggers the mixing process. The integration of deep learning models can help S predict transaction patterns based on previous activity, providing insight into potential delays and optimizing the overall mixing timeline.

3.3.2. Request acceptance

Once S receives Alice's request, it evaluates the details of the request and uses a deep learning model to predict transaction congestion, network delays, and potential bottlenecks in the processing of Alice's payment and return. By analyzing historical data and real-time network conditions, the deep learning model helps S determine the most efficient way to process Alice's transactions, minimizing delays. If S accepts the request, it responds by sending Alice the same set of parameters $(K_{in}, K_{out}, w, t_1, t_2, K_{esc}, D, D')$ along with its signature K_S to confirm acceptance. If, for any reason, S rejects the request (such as system overload or security concerns), both parties exit the protocol, ensuring that no further transactions take place without mutual agreement.

3.3.3. Payment processing

Alice must transfer the specified amount of cryptocurrency to the mixing center before the t_1 deadline. This payment is recorded on the blockchain for transparency and accountability. The deep learning algorithm embedded in the protocol monitors transaction congestion across the network and predicts the optimal timing for confirming the payment. By assessing blockchain block times and network traffic patterns, the algorithm ensures that Alice's payment is confirmed quickly, optimizing transaction processing speed. If Alice fails to make the payment by the deadline, S exits the protocol, and the transaction is voided, preventing any further actions.

3.3.4. Currency return

Upon successful payment, S is responsible for returning an equivalent amount of cryptocurrency to Alice before the t_2 deadline. This return transaction is also recorded on the blockchain for verification. The deep learning model comes into play again here by optimizing the timing and accuracy of the return process. It analyzes blockchain conditions, predicting transaction congestion and network load, to ensure that the funds are returned to Alice promptly. If Alice does not receive the return payment by the agreed-upon deadline, she can disclose the signature obtained from Step 2, thereby proving that she fulfilled her part of the protocol. This acts as a safeguard, allowing Alice to damage S's reputation if the mixing center fails to honor the transaction, thereby incentivizing S to fulfill its obligations. The incorporation of deep learning algorithms into the MixCoin protocol offers several advantages. First, it enhances the accuracy and efficiency of transaction prediction, allowing the mixing center to respond dynamically to network conditions and transaction patterns. By predicting congestion and delays, deep learning helps optimize the transaction

throughput and the overall mixing process. The model also facilitates better risk management by identifying potential failures before they occur, improving user trust and satisfaction. Additionally, the ability to dynamically adjust to changing conditions in real time ensures that the system can scale efficiently, even as the volume of transactions increases.

4. SYSTEM DESCRIPTION

The proxy-based cryptocurrency payment system is designed to enhance transaction privacy, security, and efficiency for users, with the participation of three main entities: the proxy M (a trusted intermediary), the User (individual making the payment), and the Vendor (merchant receiving the payment) [25]. The system architecture is depicted in Figure 2, and the core components of the system are organized into two main phases: the system setup phase and the deposit agreement phase. Each phase contains steps to secure communication, establish user anonymity, and ensure compliance with the protocol.

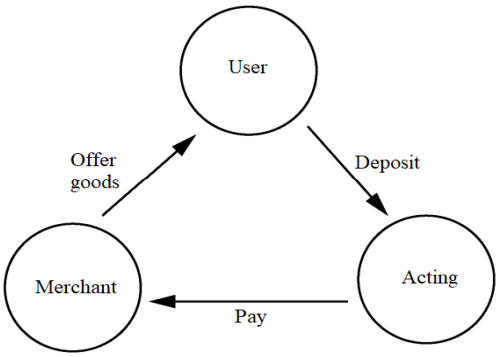

Figure 2. System structure

4.1. System setup

In the initial setup phase, each participant, including the user and the merchant, generates their own cryptographic keys, which are essential for enabling secure transactions within the system. These cryptographic keys consist of public-private key pairs, which ensure that communication and transactions are encrypted and secure. The process also involves generating one-time addresses, which are temporary public keys used to facilitate anonymous transactions, ensuring that the identities of the participants are not easily traceable. Additionally, participants implement signing protocols that protect the integrity and authenticity of transactions, preventing unauthorized modifications. This phase serves as the foundation for building a secure and private ecosystem for conducting transactions while maintaining confidentiality and trust among the parties involved.

- Proxy M key generation: (1) Long-term public key: M selects a unique long-term public key address, which will be used as a permanent point of reference for future transactions with users. (2) Public-Private Key Pair Generation: Using a partial blind signature scheme, M generates a cryptographic public-private key pair (pk, sk), where pk is the public key for verification, and sk is the private signing key. (3) Signature Key Pair: Additionally, M generates a dedicated pair of signing keys (pub_M, prv_M) to sign transactions, ensuring that all communications can be verified as authentic.
- Vendor key generation: (1) Key pairs for one-time addresses: The vendor generates two pairs of long-term public-private keys, (pka, ska) and (pkb, skb). These key pairs are used to create one-time public addresses for each payment transaction. (2) Purpose of One-Time Addresses: Each one-time public key address, generated uniquely for every transaction, provides the necessary anonymity for the User while acting as an accountable payment receipt for the vendor.
- User public key selection: In the user public key selection process, the user begins by generating and selecting a unique public key address, referred to as upk_v, specifically for deposit purposes. This address will be used by the user to deposit funds with the merchant M for future transactions, ensuring that the process remains secure and private. The upk_v public key address acts as a secure point of interaction between the user and the merchant, allowing for transactions to be conducted without exposing sensitive information. By selecting this dedicated deposit key, the user enhances their privacy, as it isolates the funds for specific transactions and minimizes the exposure of other financial details. This step is crucial for enabling trust and confidentiality in the transaction process between the user and the merchant.

4.2. Deposit agreement

The deposit agreement phase establishes a secure framework for the transaction between the User and the Merchant M, ensuring that both parties' actions are well-documented and protected. During this phase, the User initiates a deposit request by sending relevant details such as the deposit amount, their public key address, and the deadline for deposit confirmation. Upon receiving this request, the Merchant generates a commitment signature, confirming their acceptance of the deposit and sending it back to the User. This commitment, once verified, allows the User to complete the transaction on the blockchain, with the Merchant then confirming receipt of the funds by generating a final signature. This process ensures that all actions are auditable and verifiable on the blockchain, enhancing security and accountability for both the User's funds and the Merchant's obligations.

- Deposit request initiation: user deposit request: the user initiates the deposit process by sending a request to the merchant. M, containing essential transaction details. This request includes the deposit amount wt, which specifies the funds the user intends to deposit and the user's public key address upk_v, which will be used to transfer funds. Additionally, the user specifies a deadline t_1, indicating the latest time by which the merchant must acknowledge the deposit. The inclusion of this deadline ensures timely processing and provides a clear window for the merchant to confirm or reject the deposit request. The deposit request serves as the initial step in securing the transaction and triggering the subsequent stages of the deposit agreement.
- Commitment generation by M: deposit commitment: If M accepts the deposit request, it creates a deposit commitment signature $\sigma_C = Sig_M(wt, upk_v, tpk_v)$, where tpk_v represents the target public key address associated with the transaction. This signature acts as proof of commitment from M to honor the transaction. commitment transmission: M sends σ_C back to the User, which allows the User to proceed with creating the actual deposit transaction. This step ensures both parties have mutual proof of the deposit terms.
- Upon receiving the deposit commitment signature σ_C from the merchant, the user proceeds to create a transaction on the blockchain. In this transaction, the user sets their public key address upk_v as the input and the merchant's temporary public key address tpk_v as the output. The transaction specifies the deposit amount and ensures that the funds are transferred to the designated temporary address. By recording the transaction on the blockchain, the user establishes an immutable and transparent record of the deposit, ensuring that the transaction cannot be altered or tampered with. This step serves as an essential part of the deposit process, providing verifiable evidence of the user's payment for future reference.
- Upon receiving the payment from the user, the merchant M first verifies whether the transaction has been completed before the specified deadline, denoted as t_1. If the payment is confirmed within the given time frame, M generates a confirmation signature, denoted as $\sigma_M = Sig_M(upk_v, wt, T)$, where T includes a unique transaction identifier, such as a timestamp or transaction ID. This signature serves as an official record of the payment and verifies that the funds have been received correctly. Once the signature is generated, M sends σ_M to the user as proof of payment, confirming that the transaction has been successfully processed. The signature is also logged on the blockchain, providing transparency and ensuring the transaction can be publicly verified by any participant in the network.
- fallback mechanism for user protection: verification of commitment: if the User does not receive σ_M from M (indicating that M has not acknowledged the payment), the User has the right to disclose σ_C to prove compliance with the initial agreement. Public Verification: Since σ_C is a verifiable signature from M, anyone can verify the validity of σ_C on the blockchain. This mechanism provides the User with a means to safeguard their transaction, as it serves as public proof that the User fulfilled their part of the agreement.

4.3. Order submission

User's initial step: the user begins the order submission process by selecting a random value rrr within the range $[1, n-1]$.They then calculate $R = rG$, where G is a generator point on an elliptic curve, and rG is a derived point that provides cryptographic security through randomness. Using the one-time public key generation formula, the user calculates an address P for the transaction. The formula is $P = pka + H(r, v) \cdot G$, where$pka$ is the user's permanent public key, $H(r, v)$ is a cryptographic hash of r and transaction details v, which adds uniqueness. The user then sends $R \parallel msg$ to the merchant, where msg contains comprehensive order information.

Merchant's calculation: upon receiving $R \parallel msg$ from the user, the merchant calculates the transaction receipt address, represented by $P = pka + H(r, v) \cdot G$, matching the user's generated address. The merchant computer has a corresponding private key $s = ska + H(r, v) \cdot v$, which they will use for signing the transaction. Finally, the merchant generates a transaction commitment signature $\sigma_{Vender} =$

$Sig_v\ (R, msg)$ and sends it to the user, thereby confirming receipt of the order and establishing a transaction commitment.

4.4. Payment commitment

User's commitment: to secure the payment, the user blinds the one-time public key address PPP by calculating $P^* = Blind(P)$, where blinding adds a layer of security. They then create a blinded payment signature $\sigma^*_{User} = Sig_{sk_v}\ (P^*, w)$, with sk_v as the user's private signing key. The user sends $\sigma^*_{User} \parallel \sigma_M$ to the merchant. Merchant's Verification: Upon receipt, the merchant verifies the validity of σ^*_{User}. If valid, they confirm thatupk_v, the user's public key is legitimate. To optimize this process, the merchant leverages a deep learning model trained on past transaction patterns to predict validity and identify anomalies, reducing manual verification time and expediting processing. Next, the merchant checks if σ_M is unique by referencing its timestamp. If σ_M is new and the account balance is sufficient, the merchant calculates the consensus parameter $s = \tau(w)$ and generates a blinded signature $\sigma^*_{Pay} = BldSig_{sk}(s, P^*)$, incorporating the consensus parameter s.The merchant updates the balance by computing the latest account balance signature $\sigma'_M = Sg_v(upk_v, w, T)$ and sends $\sigma^*_{Pay} \parallel \sigma'_M$ to the user.

4.5. Payment agreement

User's De-Blinding: The user de-blendsσ^*_{Pay} to obtain $\sigma_{Pay}$and sends it to the merchant using an anonymous channel, such as Tor, to maintain privacy. Merchant's verification and payment processing: The merchant verifies σ_{Pay} for authenticity [26]. If valid and it is the initial use, the merchant processes the payment to the one-time public key address P, setting up a transaction with tpk_v(transaction public key for verification) as the input and P as the output, transferring the specified amount w. Advanced deep learning algorithms support the merchant by monitoring for anomalies in transaction behavior, thus verifying the authenticity of the transaction more effectively.

4.6. Transaction success

Upon observing the payment on the blockchain from the merchant to the one-time public key address P, the merchant promptly fulfills the transaction commitment, such as shipping the purchased goods, without waiting for additional block confirmations. Once the goods are received, the user utilizes the original random number r (used in generating R) as a private key to sign a receipt $\sigma_{receiver} = Sig_r(msg)$, which they send to the merchant as proof of receipt.

4.7. Accountability protocol

Addressing malicious behavior by the merchant: If the merchant fails to fulfill their commitment, the user can send the transaction commitment signature $\sigma_{Vender} = Sig_v\ (R, msg)$ to an arbitrating proxy. The proxy verifies σ_{Vender} and recalculates $P = pka + H(r, v) \cdot G$alongside$R' = rG$. If $R = R'$, it confirms that P matches the agreed-upon one-time public key address. Evidence of a transaction using P as input on the blockchain would conclusively prove the user's payment.

Addressing malicious claims by the user: If a user falsely claims that the merchant did not fulfill the commitment, the merchant can reveal $\sigma_{receiver}$, with the verification process confirming whether $\sigma_{receiver}$, proves the user's receipt of goods. Incorporating deep learning algorithms into this accountability protocol enhances the system's efficiency in detecting fraud and validating transactions. This ultimately improves the cryptocurrency payment system's reliability and performance.

5. DETAILED IMPLEMENTATION PLAN

This implementation plan outlines key steps for setting up the system and ensuring secure and private transaction handling. The process begins with selecting long-term public key addresses and generating corresponding public-private key pairs for the user, merchant, and the system. A deposit agreement is initiated by the user, who requests a deposit by sending details, including the deposit amount, the user's public key address, and a deadline [27]. The merchant then calculates a deposit commitment and sends a signature to confirm the commitment to accept the deposit. If the user receives the signature and completes the transaction on the blockchain, the merchant verifies the payment and sends a final confirmation signature. The user, in turn, can disclose the commitment signature to prove the agreement if needed. For order submission, the user calculates a random number and creates a one-time public key address, which is then used to send order information to the merchant. The system integrates cryptographic operations and deep learning techniques to ensure security, privacy, and transaction accountability throughout the process.

5.1. System setup

Long-term key selection by merchant (M): M generates a long-term public key address for receiving and verifying deposits. M selects and uses a key generation algorithm based on a partial blind signature scheme to create a pair of long-term keys: a public key pk and a private key sk. Merchant's Signing Keys: M generates a long-term signature key pair (pub_M, prv_M), where pub_M is the public signing key and prv_M is the private signing key. These keys will be used to sign transaction commitments, thereby securing transaction authenticity and providing verifiable proof of commitment. Vender's Key Pairs for generating one-time addresses: The vendor (an entity acting on behalf of M to facilitate transactions) generates two sets of long-term public-private key pairs: (pka, ska) and (pkb, skb). These key pairs are used to generate one-time public key addresses, which add privacy to the transactions by creating unique payment addresses for each user. User's deposit key selection: The user selects a public key address upk_v, which will serve as their unique deposit address for making payments to M. This address will be used to receive and verify the completion of deposit transactions, contributing to user identity privacy.

5.2. Deposit agreement

Initiating deposit request: The user initiates a deposit agreement by sending (wt, upk_v, t_1) to M, where wt is the deposit amount, upk_v is the user's deposit address, t_1 is the deadline by which the deposit must be confirmed. Merchant's deposit commitment: If M approves the deposit request, it calculates a deposit commitment signature $\sigma_C = Sig_M(wt, upk_v, tpk_v)$, where$Sig_M$denotes a signature created using $M's$ signing key, $tpk_v$is a temporary public key generated specifically for this deposit, M then sends σ_C to the user as confirmation of the commitment to accept the deposit. If M rejects the deposit is rejected, the protocol terminates here. User's payment transaction: After receiving σ_C, the user creates a transaction on the blockchain, setting upk_v as the input and tpk_v as the output, thus transferring wt to tpk_v.Merchant's Confirmation of deposit: If M receives and verifies the payment transaction before the deadline t_1, it generates a final deposit confirmation signature $\sigma_M = Sig_M(upk_v, wt, T)$, where T includes timestamp details for additional verification. M sends σ_M to the user as proof that the deposit has been accepted. Disclosure of dposit Commitment (optional): If the user does not receive σ_M from M, they may disclose σ_C as proof of their commitment. The validity of σ_C can be publicly verified, and the transaction made by the user can be traced on the blockchain, confirming the user's compliance with the agreement.

5.3. Order submission

User's random key generation: To initiate an order, the user randomly selects rrr from the range $[1, n-1]$ and calculates $R = rG$, where G is the generator point of an elliptic curve. The user then calculates a one-time public key address $P = pka + H(r, v) \cdot G$, where$pka$ is the vendor's long-term public key, $H(r, v)$ is a cryptographic hash of r and transaction-specific details v. This unique public key address P will serve as the transaction receipt address, preserving the user's anonymity.

Order information transmission: The user sends $R \parallel msg$ to the vendor, where R is the random elliptic curve point derived from rG, msg contains the order information, ensuring the vendor has the necessary details to fulfill the transaction.Vendor's Receipt Address and Commitment: Upon receiving $R \parallel msg$from the user, the vendor computes the transaction receipt address, which is $P = pka + H(r, v) \cdot G$.The vendor then calculates the corresponding private key $s = ska + H(r, v) \cdot v$, allowing them to sign the transaction. The vendor generates a transaction commitment signature $\sigma_{Vender} = Sig_v(R, msg)$ to verify receipt of the order, which they send back to the user as proof of acceptance.

5.4. Payment commitment

$User$ blinds the one-time public key address P by calculating $P^* = Blind(P)$, and computes the blinded payment information $\sigma^*_{User} = Sig_v(P^*, w)$. $User$ sends $\sigma^*_{User} \parallel \sigma_M$ to M, where sk_v is $User's$ signing private key. Upon receiving the message from $User, M$ verifies the validity of σ^*_{User}. If valid, M verifies that upk_v indeed belongs to $User$. M then checks if σ_M has been used by examining the timestamp. If σ_M is new, and the account balance is sufficient, M calculates the consensus parameter $s = \tau(w)$ and generates a blinded signature $\sigma^*_{Pay} = BldSig_{sk}(s, P^*)$, which includes the consensus parameter s. M updates the balance by computing the latest account balance signature $\sigma'_M = Sig_v(upk_v, w, T)$, and sends $\sigma^*_{Pay} \parallel \sigma'_M$ to $User$.

5.5. Payment agreement

$User$ de-blinds σ^*_{Pay} to obtain σ_{Pay} and sends it to M using an anonymous identity (e.g., Tor). M performs verification of σ_{Pay}. If valid and it is the first use, M makes the payment to the one-time public key address P: creating a transaction with tpk_v as the input and P as the output, transferring an amount w.

5.6. Transaction success

Once the merchant sees the payment from the user to the one-time public key address on the blockchain, they immediately fulfill the transaction, such as shipping the goods, without waiting for additional block confirmations. The user, upon receiving the goods, uses the random number r (used to generate the one-time public key address) as their private key to sign a message, creating a proof of receipt $\sigma_{receiver} = Sig_r(msg)$. This signature is sent to the merchant as confirmation that the goods have been received. The merchant verifies the validity of the proof using the corresponding public key, ensuring the transaction was completed as agreed. This process ensures secure and immediate transaction fulfillment, with the blockchain providing a transparent and immutable record of the transaction.

5.7. Accountability protocol

If a malicious merchant does not fulfill their commitment, $User$ sends $\sigma_{Vender} = Sig_v(R, msg)$ to M. The proxy verifies the signature. If valid, M calculates $P = pka + H(r,v) \cdot G$ and $R' = rG$. If $R = R'$, $P$is indeed the agreed-upon one-time public key. A transaction with P as input can be found on the blockchain, proving that $User$ has indeed made the payment. If a malicious user falsely claims the merchant did not fulfill their commitment, $Vender$ can disclose $\sigma_{receive}$. The signature verification algorithm will confirm if $\sigma_{receive}$ proves that $User$ has received the goods.

6. SYSTEM ANALYSIS

6.1. Validity of one-time public key address

When the user randomly selects r from the range [1, n-1], the one-time public key address P is calculated as $P = pka + H(r,v) \cdot G$, $P' = pka + H(r,v) \cdot G$. Both the user and the merchant compute the same public key address P, which is verified as $P = P'$. The corresponding private key is derived as $sa + H(r,v) \cdot b$, but only the merchant holds the private key $\boldsymbol{b_v}$ associated with the one-time public key. This ensures that only the merchant can access the private key and complete transactions. To enhance the efficiency of the key generation process, deep learning algorithms are applied, optimizing computation speed. These algorithms also improve the security of the system by ensuring quicker and stronger key generation.

6.2. Reliability

Reliability of M: The reliability of M (the mixing center) is a key aspect of the system, and it refers to the user's ability to prove M's dishonesty if M engages in fraudulent behavior. In the context of this system, the reliability of M is safeguarded by a mechanism that allows users to disclose M's signature and associated transaction details on the blockchain should M behave dishonestly. This transparent and traceable process enhances the trustworthiness of the mixing center. To further strengthen the reliability of the system, deep learning algorithms are employed to continuously monitor transaction patterns and detect any anomalous behavior from M. These algorithms analyze blockchain transactions for signs of inconsistency or fraud, such as suspicious timing, unusually large transactions, or deviations from normal behavior. By identifying such anomalies in real time, deep learning models can help prevent dishonest actions from M, ensuring that the system remains reliable and secure for users.

Non-forgery: The non-forgery property of the signature algorithm is crucial for ensuring the integrity and authenticity of transactions. This property guarantees that once a signature is generated for a transaction, the user cannot alter the transaction details—such as the account balance or the payment information—without invalidating the signature. Essentially, only the holder of the private key associated with the public key used in the transaction can generate a valid signature that includes payment details. To further protect against fraud, even if an attacker manages to steal a legitimate signature with payment details and attempts to submit it to M, the system ensures that M will only transfer the corresponding cryptocurrency to the public key address included in the signature.

Since the attacker does not possess the corresponding private key, they will be unable to use or access the cryptocurrency, rendering the stolen signature useless. To enhance this non-forgery property, deep learning techniques are integrated into the system to detect and prevent counterfeit signatures. These algorithms analyze patterns in signatures and transactions to identify potential forgeries. By training on large datasets of legitimate signatures and transactions, deep learning models can spot subtle discrepancies or anomalies that may indicate a forged signature. This proactive approach to signature validation strengthens the overall security of the system, ensuring that only authorized users can perform legitimate transactions and preventing malicious actors from exploiting vulnerabilities in the signature process.

6.3. Anonymity

Passive attacks: The system is designed to be resistant to passive attacks, ensuring that attackers cannot passively monitor or intercept users' information without detection. In line with methods used in previous research [17], the system prevents passive attackers from linking a user's deposit account to the one-time public key address used for transactions, effectively safeguarding additional payment privacy. The system's anonymity level improves as more honest users engage with the mixing service (denoted as M), as the larger number of participants increases the complexity of tracking and correlating transactions. This growth in participants directly correlates with M's available resources, making the task of linking transactions more difficult. To enhance this protection, deep learning models are integrated into the system. These models continuously monitor user behaviors and transaction patterns to detect any unusual activity that might indicate attempts to breach anonymity. By analyzing trends and emerging threats, these models improve the system's ability to mitigate passive threats, enhancing the overall privacy protection for users.

Active attacks: While the system is robust against passive attacks, it cannot fully defend against active attacks. Active attackers may attempt to manipulate the network to reduce the perceived number of anonymous users, making it easier for them to identify individual participants. These attackers may deploy various network-based attacks [20] to disrupt or obscure the anonymity of users. To combat such threats, the system incorporates deep learning-based anomaly detection techniques that analyze network traffic and user behavior for signs of manipulation or malicious activity. By using these advanced algorithms, the system increases the cost and complexity of executing successful active attacks. The deep learning models can identify irregular patterns and flag potential attacks in real time, raising the difficulty for adversaries trying to break the anonymity protocol.

Attacks by M: Although the system is well-protected against attacks from passive and active attackers, it can also defend against attacks originating from the mixing center (M). The partial blind signature algorithm plays a critical role in maintaining user anonymity by ensuring that M cannot link a user's deposit address with the one-time public key address used for the transaction. Moreover, M does not have any knowledge of which merchant the one-time address belongs to, preserving transaction confidentiality. However, the system is still vulnerable to timing attacks, where M might deduce the user's identity by analyzing the timing of transactions. For instance, if M signs the user's blinded payment information when the number of users is very low and the user then quickly sends the de-blinded payment information back to M, there is a higher likelihood that M could link the user to the one-time public key address. To address this vulnerability, advanced deep learning algorithms are implemented to detect and mitigate timing attacks. These algorithms monitor transaction timing and sequence patterns, identifying potential risks associated with rapid de-blinding and sending of payment information. By analyzing transaction behaviors, the deep learning models can flag suspicious activities that may indicate an attempt to exploit the timing gap. This proactive approach enables the system to detect and counteract timing attacks more effectively, ensuring that the anonymity of users is upheld.

6.4. Effectiveness

Fairness: The system is designed to ensure fairness for all parties involved in the transaction. From the merchant's standpoint, payment confirmation is a prerequisite for shipping goods, thus protecting their interests by guaranteeing that no goods are shipped without proper payment. On the other hand, from the user's perspective, once the payment is completed and confirmed, the merchant is unable to deny the transaction, ensuring that the user is not left at a disadvantage. This mutual assurance fosters trust between both parties and promotes fairness in the transaction process.

Timeliness: Initialization phase: The time-consuming processes in this phase mainly involve the execution of the signature algorithm, transaction creation, and the initial confirmation. However, these steps need to be performed only once during the initialization of the system, which does not have a significant impact on the time required for subsequent payments. Transaction start phase: This phase involves two key steps: executing the signature algorithm and creating the transaction. While these operations consume time, the overall time required is minimal, thanks to the optimized execution of the signature algorithm. Transaction completion phase: The final phase mainly focuses on generating the receipt signature, which is necessary for confirming the transaction's completion. Though this step contributes to some time consumption, it is streamlined for efficiency. Overall, the system employs the signature algorithm three times per transaction. By leveraging efficient, short signature algorithms and incorporating deep learning techniques for optimizing signature processing, the system significantly reduces the time required for transaction confirmation. In comparison to traditional systems, which often require up to 60 minutes for confirmation, this approach enhances the overall transaction speed, improving efficiency and reducing delays.

Compatibility: One of the key strengths of this system is its ability to integrate with existing blockchain technologies without altering the underlying protocol. This ensures that the new system is fully compatible with current blockchain systems, allowing for easy adoption and integration. The addition of deep learning algorithms further enhances the system's performance by optimizing processes such as transaction verification, signature generation, and network congestion management, all while maintaining full compatibility with existing protocols. This compatibility makes it possible to adopt these advanced features without disrupting the overall blockchain ecosystem.

7. CONCLUSION

The decentralized structure of blockchain technology, while offering significant advantages such as enhanced security and transparency, also presents several challenges that hinder the practical adoption of blockchain-based cryptocurrencies. Among these challenges are payment delays and potential privacy vulnerabilities, which can complicate the seamless execution of transactions. This paper proposes a solution to address these issues by introducing a proxy as a payment intermediary, along with the integration of advanced deep learning algorithms to further optimize the system's performance. The proxy acts as an intermediary, reducing the time required to confirm transactions and improving user privacy by obfuscating direct links between transaction parties. This approach not only alleviates payment delays but also enhances the overall efficiency of the blockchain-based payment system. Importantly, the use of the proxy does not compromise the decentralized or deflationary nature of the blockchain, ensuring that currency generation remains consistent with its original principles.

By leveraging deep learning models, the system is able to dynamically optimize various aspects of the payment process, including predicting network congestion, reducing transaction confirmation times, and mitigating potential privacy breaches. These innovations result in a more efficient, secure, and privacy-preserving cryptocurrency payment system, making it more viable for widespread adoption in practical applications.

FUNDING INFORMATION

Authors state no funding involved.

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] C.-X. Wang *et al.*, "On the road to 6G: Visions, requirements, key technologies, and testbeds," *IEEE Commun. Surv. Tutorials*, vol. 25, no. 2, pp. 905-974, 2023, doi: 10.1109/COMST.2023.3249835.

[2] Gervais, S. Capkun, G. O. Karame, and D. Gruber, "Privacy considerations for lightweight bitcoin clients using bloom filters," in *Proc. Annu. Computer Security Applications Conf.*, 2014, pp. 326-335, doi: 10.1145/2664243.2664267.

[3] J. Bonneau *et al.*, "SoK: research perspectives and challenges for bitcoin and cryptocurrencies," in *IEEE Symp. Security and Privacy*, 2015, pp. 104-121, doi: 10.1109/SP.2015.14.

[4] J. Andrew *et al.*, "Blockchain for healthcare systems: Architecture, security challenges, trends and future directions," *J. Network and Computer Applications*, vol. 215, p. 103633, 2023, doi: 10.1016/j.jnca.2023.103633.

[5] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in Bitcoin," in *Proc. ACM Conf. Computer and Communications Security*, 2012, pp. 906-917, doi: 10.1145/3641546.

[6] Eyal and E. G. Sirer, "Majority is not enough: bitcoin mining is vulnerable," in *Proc. Int. Conf. Financial Cryptography and Data Security, Springer*, 2014, doi: 10.1145/3212998.

[7] P. Koshy, P. Koshy, and P. McDaniel, "An analysis of anonymity in bitcoin using P2P network traffic," in *Proc. Int. Conf. Financial Cryptography and Data Security*, 2014, pp. 469-485, doi: 10.1007/978-3-662-45472-5_30.

[8] S. Meiklejohn *et al.*, "A Fistful of Bitcoins: characterizing payments among men with no names," in *Proc. ACM Internet Measurement Conf.*, 2013, pp. 127-140, doi: 10.1145/2504730.2504747.

[9] F. Reid and M. Harrigan, "An analysis of anonymity in the bitcoin system," in *Security and Privacy in Social Networks, Springe*r, 2012, pp. 197-223, doi: 10.1007/978-1-4614-4139-7_10.

[10] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic digital signature algorithm (ECDSA)," *Int. J. Information Security*, vol. 1, no. 1, pp. 36-63, 2010, doi: 10.1007/s102070100002.

[11] H. Hu *et al.*, "A practical anonymous voting scheme based on blockchain for Internet of energy," *Security and Communication Networks*, vol. 2022, no. 1, p. 4436824, 2022, doi: 10.1155/2022/4436824.

[12] M. Natarajan *et al.*, "Quantum secure patient login credential system using blockchain for electronic health record sharing framework," *Scientific Reports*, vol. 15, no. 1, p. 4023, 2025, doi: 10.1038/s41598-025-86658-9.
[13] M. Abe and E. Fujisaki, "How to date blind signatures," in *Advances in Cryptology*, *Springer*, 1996, pp. 244-251, doi: 10.1007/BFb0034851.
[14] Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: anonymous distributed e-cash from bitcoin," in *IEEE Symp. Security and Privacy*, 2013, pp. 397-411, doi: 10.1109/SP.2013.34.
[15] E. Ben-Sasson *et al.*, "Zerocash: decentralized anonymous payments from bitcoin," in *IEEE Symp. Security and Privacy*, 2014, pp. 459-474, doi: 10.1109/SP.2014.36.
[16] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "CoinShuffle: practical decentralized coin mixing for bitcoin," in *Computer Security – ESORICS*, *Springer*, 2014, pp. 345-364, doi: 10.1007/978-3-319-11212-1_20.
[17] Bonneau *et al.*, "MixCoin: anonymity for bitcoin with accountable mixes," in *Proc. Int. Conf. Financial Cryptography and Data Security*, 2014, pp. 486-504, doi: 10.1007/978-3-662-45472-5_31.
[18] Valenta and B. Rowan, "Blindcoin: blinded, accountable mixes for bitcoin," in *Proc. Int. Conf. Financial Cryptography and Data Security*, 2015, pp. 112-126, doi: 10.1007/978-3-662-48051-9_9.
[19] D. Berger, M. Lemoudden, and W. J. Buchanan, "Post-quantum migration of the tor application," *J. Cybersecurity and Privacy*, vol. 5, no. 2, p. 13, 2025, doi: 10.3390/jcp5020013.
[20] R. Douceur, "The Sybil Attack," in *Proc. First Int. Workshop Peer-to-Peer Systems*, 2002, pp. 251-260, doi: 10.1007/3-540-45748-8_24.
[21] C. Smith and A. Kumar, "Crypto-Currencies–An introduction to not-so-funny moneys," in *Contemporary Topics in Finance: A Collection of Literature Surveys*, 2019, pp. 351-381, doi: 10.1002/9781119565178.ch12.
[22] Zhu *et al.*, "Blockchain-based digital asset circulation: a survey and future Challenges," *Symmetry*, vol. 16, no. 10, p. 1287, 2024, doi: 10.3390/sym16101287.
[23] Z. Gu and O. Dib, "Enhancing fraud detection in the Ethereum blockchain using ensemble learning," *PeerJ Computer Science*, vol. 11, p. e2716, 2025, doi: 10.7717/peerj-cs.2716.
[24] Marouan *et al.*, "Empowering education: leveraging blockchain for secure credentials and lifelong learning," in *Blockchain Transformations: Navigating the Decentralized Protocols Era*, *Cham: Springer Nature Switzerland*, 2024, pp. 1-14, doi: 10.1007/978-3-031-49593-9_1.
[25] X. Xu *et al.*, "Machine learning and zero knowledge empowered trustworthy bitcoin mixing for Next-G consumer Electronics Payment," *IEEE Trans. Consumer Electronics*, 2024, doi: 10.1109/TCE.2024.3361690.
[26] H. Xie *et al.*, "SofitMix: A secure offchain-supported bitcoin-compatible mixing protocol," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 5, pp. 4311-4324, 2022, doi: 10.1109/TDSC.2022.3213824.
[27] Dandotiya *et al.*, "A deep perspective of blockchain applications in the healthcare sector and Industry 4.0," in *Artificial Intelligence in Biomedical and Modern Healthcare Informatics*, Academic Press, 2025, pp. 31-43, doi: 10.1016/B978-0-443-21870-5.00004-2.

BIOGRAPHIES OF AUTHORS

Vinay Kumar Kasula has a master's in technology from Osmania University, followed by a second Master's from Northwest Missouri State University. A deep-rooted interest in research and advanced technologies has driven his academic journey. This passion led him to pursue and complete a Ph.D. at the University of the Cumberlands, where he graduated in 2024, focusing on areas including blockchain, artificial intelligence, deep learning, machine learning, data analytics, and cybersecurity. With over a decade of industry experience, currently as a senior systems analyst, he has applied and expanded his expertise in these domains, continually exploring and integrating innovative solutions to meet evolving technological challenges. This blend of education, research, and practical experience fueled his commitment to advancements in these cutting-edge fields. He can be contacted at email: vinaykasula.phd@ieee.org.

Dr. Akhila Reddy Yadulla holds an MS in Computer Science from Northwestern Polytechnic University in 2016, an MS in Information Technology Management from Campbellsville University in 2019, and a Ph.D. in Information Technology from the University of the Cumberlands in 2024. She is an accomplished senior software engineer with over eight years of experience in Java full-stack development and expertise in the software development life cycle. She is proficient in Java, Spring Boot, Hibernate, RESTful APIs, Angular, React, Microservices, SQL/NoSQL databases, DevOps, and agile methodologies. Her research focuses on cybersecurity, artificial intelligence, data science, big data, blockchain technologies, amazon web services, and machine learning. She combines deep technical knowledge, advanced problem-solving skills, and AI-driven innovative solutions in cutting-edge technologies. Notably, she led a team in developing a secure and scalable microservices architecture for significant telecommunication and banking platforms, resulting in a major increase in system performance. She can be contacted at email: akhilareddyyadulla@ieee.org.

Bhargavi Konda works as a senior systems analyst in the human resources information systems department for a healthcare company. She mainly works with data analysis and generates reports for various departments in the organization. She has been one of the major contributors in maintaining employee records, supporting all HR departments, and especially working with payroll operations from her team. She earned her master's degree from Osmania University (2001-2004). She received the gold medal and star student of the year awards for her outstanding performance. She was placed in an IT consulting company right after her master's, and she worked as an Oracle technical consultant from 2004 to 2012. In that period, she was awarded the Outstanding Trainer award for her commitment and dedication. She has experience implementing applications in the banking and manufacturing sectors. She was passionate about higher studies, so she started to pursue a Ph.D. in IT in 2019. She earned a Ph.D. degree in Information Technology in the year 2024 from the University of the Cumberlands. She is excited to work on many journals and publications in the future and wants to continue her research. She can be contacted at email: bhargavikonda@ieee.org.

Mounica Yenugula is a skilled IT professional with a strong academic background and an experience Cloud Engineer. She holds a Ph.D. in Information Technology from the University of Cumberlands, where her research focused on critical security and trust issues in cloud environments. Her academic journey began with a Bachelor of Technology in Electronics and Communication Engineering from Koneru Lakshmaih University and continued with a Master of Science in Electrical Engineering from California State University, Los Angeles. As a certified Cloud Engineer, Mounica has successfully implemented cloud solutions within the financial sector, driving digital transformation. Her active involvement in the women in IT community and her role as a judge for the Globee Awards demonstrate her commitment to industry advancement and innovation. She can be contacted at email: ymounica.phd@ieee.org.

SuprajaAyyamgari is currently a Ph.D. candidate in Information Technology at the University of the Cumberlands, USA. She holds a robust academic foundation with a bachelor's degree in computer science from Jawaharlal Nehru University, India, a Master's degree in Information Technology from Southern New Hampshire University, USA, and an MBA from New England College, USA. Alongside her academic accomplishments, Supraja has nearly eight years of diverse industry experience in IT, where she has held roles such as Developer, Full Stack Engineer, and DevOps Specialist. She now leads critical infrastructure and cloud transformation projects in the financial sector. Her research interests include blockchain, cryptography, cloud computing, augmented reality, and virtual reality. Supraja is dedicated to advancing her expertise and is eager to contribute to academic publications and journals in the future. She can be contacted at email: sayyamgari32587@ucumberlands.edu.