

CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance

Kamalika Bhowal ^{1,✉}, Srijit Mondal ¹, Kousik Dasgupta ², Abir Chattopadhyay ³

1. *Computer Science and Technology, Institute of Engineering and Management (IEM), University of Engineering and Management, Kolkata, IND*

2. *Computer Science and Engineering, Kalyani Government Engineering College, Kalyani, IND*

3. *Electronics and Communication Engineering, Institute of Engineering and Management (IEM), University of Engineering and Management, Kolkata, IND*

Received: February 27, 2026 | Review began: March 23, 2026 | Review ended: April 02, 2026 | Published: April 09, 2026

© Copyright 2026

This is an open access article distributed under the terms of the Creative Commons Attribution License CC-BY 4.0., which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract

In decentralized finance (DeFi), accidental cryptocurrency transfers to incorrect wallet addresses are a large usability and security issue, frequently causing permanent loss of funds. We present CryptoSafeSend, a smart contract-based safety protocol for transactions featuring a cryptographically secure passcode verification scheme supporting conditional transfers. This work addresses higher-level security issues by introducing a PBKDF2-based key derivation function, which generates strong encryption keys based on Firebase's Firebase Unique Identifier. Secret passcodes are encrypted in Advanced Encryption Standard operating in Galois/Counter Mode functions deployed underneath the Web Crypto API, and the initialization vector and ciphertext are Base64 encoded for off-chain reliable storage and Firebase Firestore-based messaging. The protocol prevents unauthorized exploitation by safely binding off-chain passcode transmission to a matching on-chain verification, reinforcing user trust without undermining decentralization. Also, to ensure recoverability, CryptoSafeSend integrates a 7-day on-chain escrow lock, after which unclaimed funds become permissionlessly refundable to the sender, preserving decentralization while eliminating reliance on trusted intermediaries. Testing on an Ethereum testnet confirms negligible gas overhead, immunity against double claims, and strong security guarantees, qualifying CryptoSafeSend as a valuable constituent in next-generation secure digital asset protocols.

Categories: Blockchain and Cryptocurrency, Computational Science and Engineering, Cryptanalysis

Keywords: secure transactions, smart contracts, ethereum, passcode-protected transfers, decentralized finance (defi)

Introduction

Cryptocurrencies such as those based on blockchain technology allow users to transfer digital assets in a decentralized manner without using any financial intermediary [1]. However, some usability issues still exist in terms of cryptocurrency transactions. Among such issues, one of the most important problems occurs in situations where users accidentally send their cryptocurrencies to incorrect wallet addresses. Once such transactions are confirmed on the blockchain, they are generally irreversible [2]. According to some previous studies, user behavior and wallet handling practices significantly impact the security of cryptocurrency transactions. For example, studies by Fröhlich et al. [3,4], along with Yu et al. [5], show that users often experience difficulties in verifying wallet addresses and understanding blockchain workflows. These difficulties increase the probability of operational errors in cryptocurrency transactions, resulting in the loss of digital assets. In Figure 7, the steps involved in cryptocurrency transactions are illustrated. However, some researchers

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI https://doi.org/10.7759/s44389-026-00054-6 Page 1 of 37

have proposed various mechanisms to improve transaction authorization and ensure the security of blockchain-based financial transactions. For instance, Hayama et al. [6] introduced a recipient authorization mechanism in which users must receive approval from the recipient party for transaction completion. Similarly, Kamidoi et al. [7] proposed a protocol based on blockchain technology in which users must receive acknowledgment from the recipient party for transaction completion.

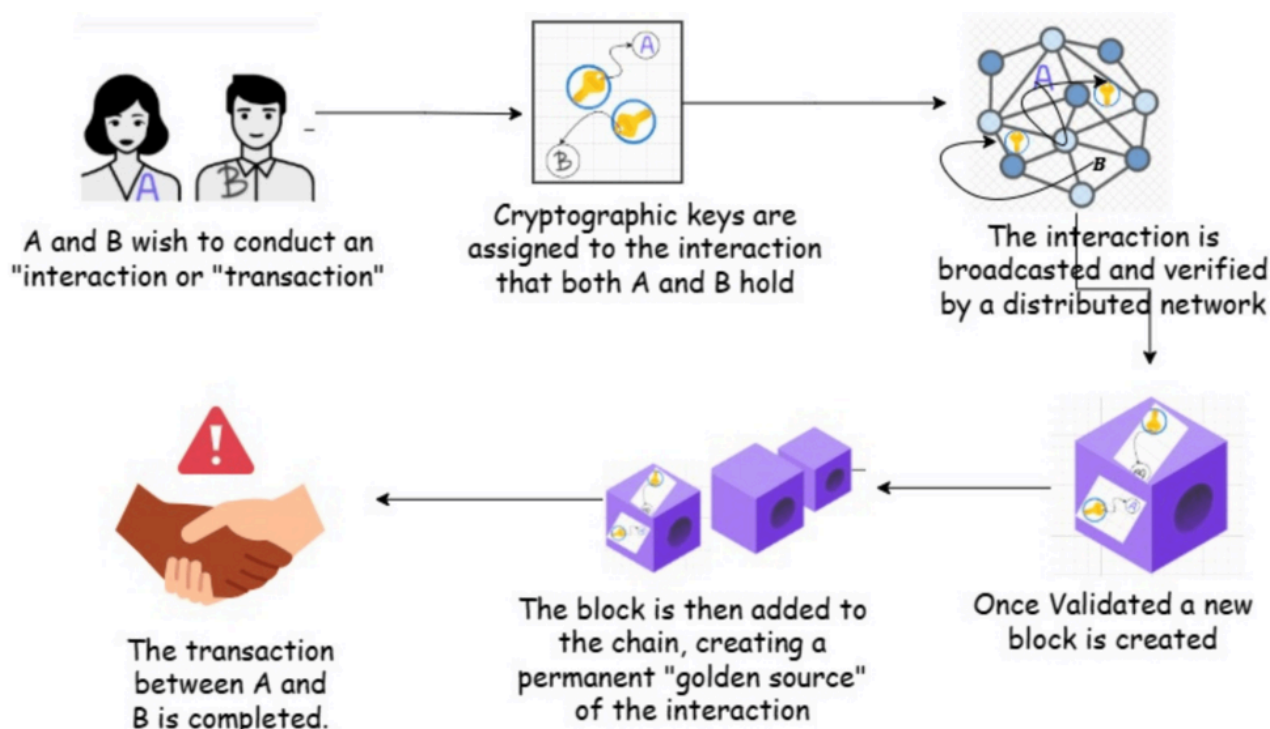


FIGURE 1: Steps Involved in a Cryptocurrency Transaction.

Source: Author's own work.

A major limitation associated with most of the existing blockchain technology is that they tend to verify the sender's cryptographic information instead of validating the accuracy of the recipient's address. Most traditional banking systems carry out various validation checks for the sender and recipient information before initiating a particular transaction. Blockchain technology, however, accepts even syntactically correct addresses without validating whether they belong to the intended recipient. As discussed in various studies, this limitation may cause users to confuse wallet addresses with token contract addresses or send transactions to incorrect networks, as highlighted in studies on token standards and smart contract behavior [8-10]. Such situations may often result in the loss of assets, which is irreversible in most cases [2].

To overcome these challenges, various techniques have been proposed to enhance the validation and verification of transactions. For instance, Krishna et al. [11] have used blockchain technology to implement client-assisted verification techniques to enhance the transparency and reliability of transactions. Nevertheless, even with the implementation of various technological barriers, users may still not be sufficiently protected against various risks associated with cryptocurrency transactions.

Besides focusing on technological barriers, researchers have also identified various behavioral and psychological factors that may affect the overall safety and security of cryptocurrency transactions. According to various studies conducted by Abramova et al. [12], Mai et al. [13], and Guan et al. [14], users may not possess adequate technical knowledge and may

How to cite this article:

have misconceptions about the working of the blockchain technology, thus leading to risky transactions. Furthermore, various investigations have revealed that users may not have adequate security practices and may not be sufficiently familiar with the wallets, which may contribute to most of the risks associated with transactions [3].

Security issues in the cryptocurrency environment have also been found to be beyond just usability. There have been various studies conducted on fraud patterns and attacks that have been directed at blockchain technology users. For example, studies have been conducted by Islam et al. [15], Soni et al. [16], and Zamani et al. [17], focusing on various fraud patterns, including cryptocurrencies. In addition, studies have been conducted on transaction traceability, indicating that blockchain technology transaction information can be used to identify suspicious activities across multiple ledgers [18].

From a broader perspective, the structural evolution of cryptocurrency address structures has also been examined to understand the implications of such structures on security and interoperability. For instance, Chan et al. [19] examined the evolution of Bitcoin address structures from a security perspective. Moreover, Omote et al. [20] suggested methods to assign trust values to addresses of the blockchain to ensure improved reliability before executing transactions. Usability-oriented research has also emphasized the importance of user-centric design guidelines while developing decentralized systems. In this regard, Froehlich et al. [21] conducted a systematic literature review to identify significant usability and human-computer interaction issues related to blockchain technology.

Technological solutions have been proposed to address issues of transaction errors and wallet security. For instance, hardware-based solutions, QR code-based verification methods, and confirmation mechanisms to ensure accurate user input are being considered [22]. Additionally, wallet recovery methods are being examined to assist users in accessing inaccessible cryptocurrency wallets [23]. Nonetheless, user-level vulnerabilities are one of the most significant yet under-addressed issues within decentralized financial systems [16,17].

The users can access the blockchain networks via cryptocurrency wallets, which store the cryptographic keys instead of the cryptocurrencies. There are different types of cryptocurrency wallets, which can be categorized into different types based on their internet connectivity and cryptographic key management. Hot wallets are those that have internet connectivity, which allows users to access their accounts via mobile applications or browser extensions. Cold wallets store the private keys offline, which makes them more secure [22].

The underlying technology behind the blockchain has undergone significant changes and improvements since the launch of Bitcoin in 2008. Initially, it was used to allow peer-to-peer transactions of digital currencies. However, with the development of smart contracts, it has been used to build decentralized applications, decentralized finance, and tokenized digital currencies [24]. A typical transaction in the cryptocurrency market starts with the user initiating the transaction via the application, which is then authorized by the user using their private key. After authorization, the transaction is broadcast to the blockchain network, which verifies the digital signature, balances, and protocol before adding the transaction to the blockchain. Depending on the type of blockchain network, the transaction is verified by the validators or miners before it is added to the distributed ledger, as shown in Figure 7. This ensures the transparency and immutability of the data [24]. However, as discussed in the study by Alruwaili and Kruger [25], it is essential to incorporate user-centered threat modeling in the cryptocurrency market. Further improvements have been made to the underlying technology behind the blockchain, which has been used to build decentralized finance and tokenized digital currencies [26].

Despite the development of advanced technology in the field of cryptocurrencies, the blockchain networks have not yet been able to implement measures that can restrict the users from sending cryptocurrencies to wrong or unintended users. Solutions like multi-signature wallets and smart contract-based custody have been implemented, which have been used to manage assets in organizations. This has created more complexity in the operation of the organizations, which may not be appropriate for conducting peer-to-peer transactions. A study by Alruwaili and Kruger [25] emphasized the importance of user-centered threat modeling, while Froehlich et al. [27] further highlighted the role of user-related security threats in cryptocurrency systems.

How to cite this article:

To overcome these limitations, this study proposes a Web3-based transaction mediation framework called CryptoSafeSend, which is aimed at improving transaction security using a double-verification process. The proposed framework presents a passcode-based escrow process in which funds are temporarily locked in a smart contract and made available only after receipt of a shared secret passcode from the recipient of the transaction. The proposed framework aims to minimize transaction risks while maintaining the decentralized nature of blockchain-based systems.

The proposed framework leverages browser-based cryptographic libraries to securely generate passcodes and conduct encrypted communications, while smart contracts running on the Ethereum blockchain manage escrow-based transaction validation and enforcement. Moreover, CryptoSafeSend proposes a time-based escrow account recovery process in which unclaimed transaction amounts are automatically returned to the sender after a specified period of time.

Experimental evaluation of the proposed framework using an Ethereum test network shows that it incurs minimal gas costs while successfully preventing double-claim attempts and enforcing secure transaction conditions. The proposed CryptoSafeSend framework combines cryptographic verification, transaction mediation using escrow accounts, and user-centric transaction safety features to provide a lightweight yet powerful solution for improving transaction reliability in decentralized digital asset systems.

Materials And Methods

Cryptocurrency is kept in a digital wallet, which can be online, on a computer, or on an external hard drive. A wallet address, which is typically a lengthy string of numbers and letters, is a feature of a digital wallet. Crypto-related crimes continue to pose a significant risk to the digital asset sector. The first half of 2025 saw a large increase in stolen funds and scams, with threat actors targeting both cryptocurrency exchanges and individual wallets with sophisticated approaches. According to blockchain analytics firm Chainalysis, over \$2.17 billion has already been stolen from cryptocurrency services this year, surpassing the total amount lost in the whole of 2024. We cannot recover our funds if such incidents happen to our wallet or cryptocurrency funds. There are other incidents, such as online exchange platforms closing, sending cryptocurrency to the wrong person, and forgetting the password of our digital wallet, which also is a threat. Because cryptocurrencies are exclusively available online, they differ significantly from conventional currencies like U.S. dollars [4]. The CryptoSafeSend platform is designed to enhance security in peer-to-peer cryptocurrency transfers by combining off-chain user verification with on-chain transaction control. This hybrid model ensures that users can only claim funds if they meet predefined security criteria, such as entering a valid passcode and completing identity verification. The overall methodology encompasses frontend authentication, user-friendly onboarding, off-chain identity management, and secure on-chain contract interaction.

Smart contract operations

The smart contract is deployed on the Ethereum blockchain, which helped us to achieve our desired functionality, Create Escrow, which is the createTransaction function for CryptoSafeSend, as shown in Appendices Figure 23, making it easier to create an escrow. By supplying the recipient's address, the hashed passcode, and the ether amount to be escrowed, this feature enables a sender to start a secure transfer. The transaction has a unique ID and is initialized with unclaimed and uncanceled states before being recorded on-chain in a mapping. The function ensures transparent and traceable fund locking under smart contract control by emitting a TransactionCreated event and including essential validation steps.

The claim function enables the recipient to securely retrieve escrowed Ether by providing the correct plaintext passcode and the corresponding escrow transaction identifier, as shown in Appendices Figure 24. The smart contract first verifies that the caller is the intended recipient and confirms that the escrow has neither been claimed nor refunded. During transaction creation, a bytes32 passcode hash is generated using the Keccak-256 cryptographic hash function and stored within the transaction record. During the claim process, the recipient submits the plaintext passcode, which is re-hashed by the contract and compared against the stored hash value. Upon successful verification, the contract transfers the

How to cite this article:

escrowed Ether to the authorized recipient, updates the transaction status to claimed, and emits a corresponding event to ensure transparency and auditability. This mechanism ensures that only the intended recipient possessing the correct passcode can access the locked funds, thereby enhancing transaction security and preventing unauthorized claims.

The refund mechanism enables the sender to recover escrowed funds if the recipient fails to claim the transaction within the predefined escrow period. The smart contract implements a time-based validation using the blockchain timestamp to ensure that refunds can only be executed after the escrow expiration time has elapsed, as shown in Appendices Figure 25. Specifically, the contract verifies that the transaction has not already been completed and confirms that the current block timestamp exceeds the stored expiry time. These validation checks prevent premature or unauthorized refunds. Upon successful verification, the contract marks the transaction as completed and securely transfers the escrowed Ether back to the sender's wallet. Additionally, a TransactionRefunded event is emitted to ensure transparency, traceability, and auditability of the refund process. This time-locked escrow mechanism ensures decentralized fund recovery without relying on trusted intermediaries while preserving the security and integrity of the transaction lifecycle.

The viewing functionality, as in Appendices Figure 26, allows users to retrieve and monitor the details of escrow transactions associated with their account. By entering the unique transaction identifier, users can access relevant transaction information, including the sender address, recipient address, escrow amount, and current transaction status. This feature enhances transparency by enabling users to track the progress of their escrow transactions and verify whether funds have been claimed, refunded, or remain locked. The viewing mechanism supports user awareness and accountability while ensuring easy access to transaction records stored securely on the blockchain.

User authentication and onboarding

User verification is a vital step to confirm the legitimacy of users. To confirm a user's identity and provide access to customized services, information, or features, sign-in is necessary. It enables systems to differentiate users, implement access control, and guarantee that only authorized users can access sensitive data. Offering user-specific features like dashboards, saved preferences, or transaction history would be impossible without a sign-in mechanism. Logging and auditing, which are essential for keeping an eye on user behavior and upholding compliance in regulated environments, are also made possible by sign-in. When we launch the CryptoSafeSend WebApp, the user is authenticated and onboarded by several user preferences while also ensuring a robust level of security. Google Authentication uses contemporary protocols like OAuth 2.0 and OpenID Connect, which enable apps to authenticate users without directly handling passwords. It is regarded as extremely secure. With features like multi-factor authentication, suspicious login detection, and advanced threat prevention driven by Google's infrastructure, it provides robust protection. Users in CryptoSafeSend can authenticate with Google Auth through Firebase (Sign in with Civic) or a basic email/password setup, with the system itself enabling multimodal capabilities when it comes to login. Vite, React, and TypeScript are used to enable the email/password option, through which user credentials are safely hashed and stored. Accessibility through the flexible auth model is maintained for common users, as users are more accustomed to decentralized identity solutions.

The registration and login pages demonstrate how the user interface of CryptoSafeSend ensures a quick and easy process that guides the user through the authentication process with minimum effort. Firebase automates user account management, establishes sessions, and verifies credentials behind the scenes following a choice of the login method. Upon the success of the authentication process, the users are directed to the dashboard, and they can use the native smart contract abilities of the platform to start safe transactions, as shown in Figure 2. Effective and secure entry into the dApp space is guaranteed through such an optimized onboarding process. React and TypeScript are used throughout the frontend building process, and they provide a new, responsive, and interactive design. Session security, as well as user authentication, is taken care of entirely by Firebase. With the two-interface design provided by the platform, the user can register a new account or use their current login information. Figure 3 shows the user registration form, and Figure 4 shows the email/password UI for the log-in. After verification, users are allowed access to the transaction UI, through

How to cite this article:

which they can set up a secure transfer of funds capped with a hashed passcode. With such an authentication process, only proven users have interactions with smart contracts on the blockchain, minimizing the risk of unauthorized use or misuse through phishing. The user can also sign in with Google Authentication, as shown in Figure 5.

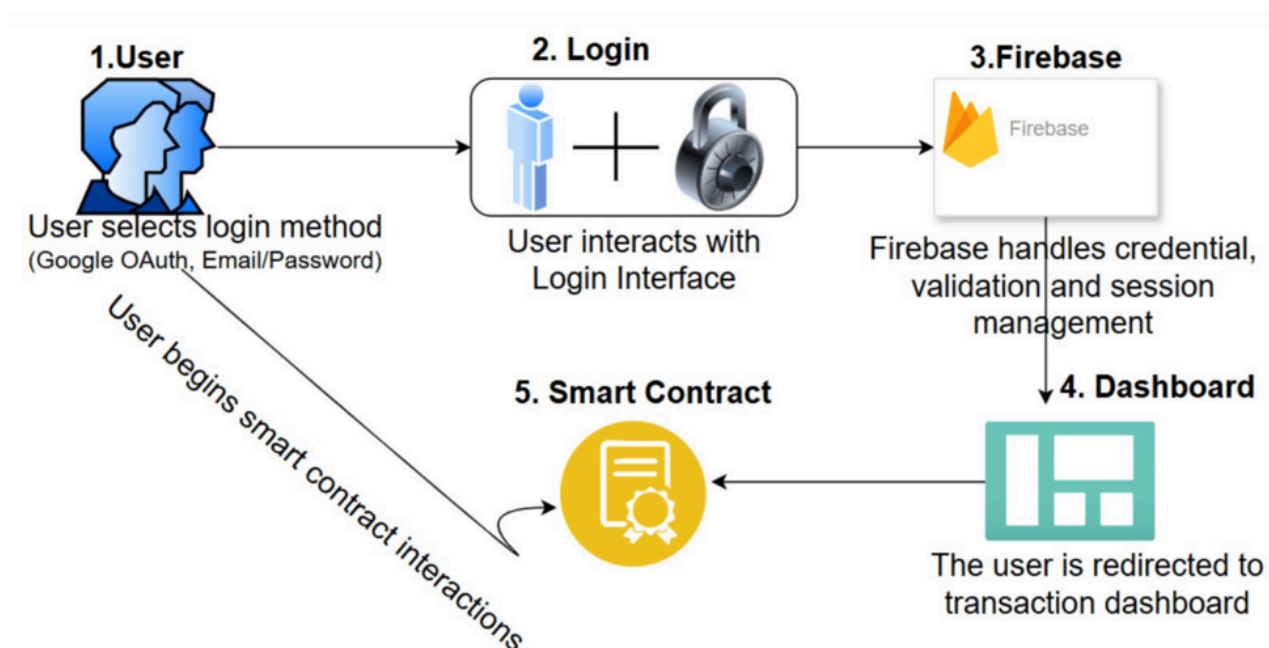
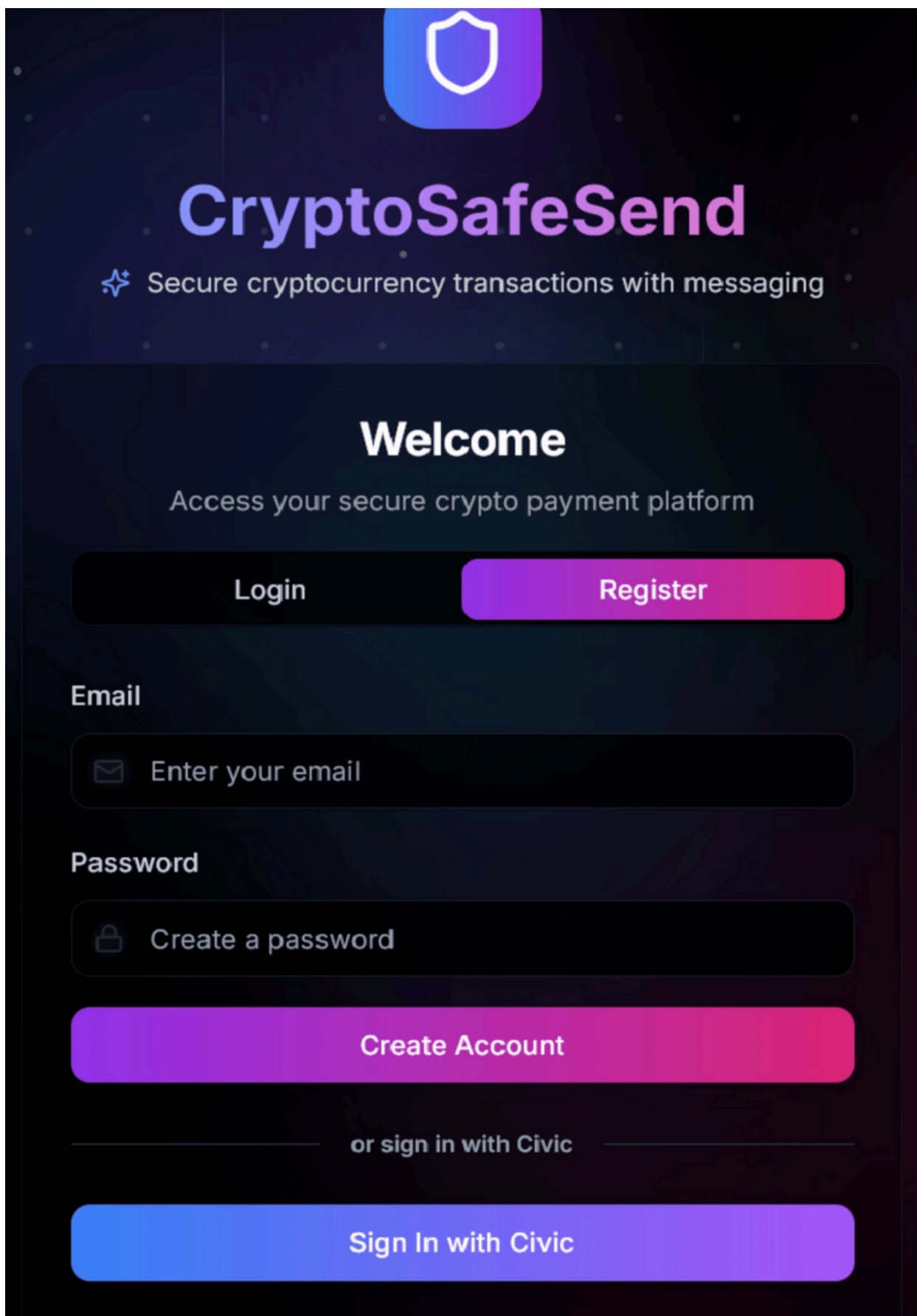


FIGURE 2: Authentication Flow.

Source: Author's own work.



The image shows a digital interface for 'CryptoSafeSend'. At the top, there is a shield icon inside a purple-to-blue gradient rounded square. Below this, the text 'CryptoSafeSend' is written in a large, bold, purple-to-blue gradient font. Underneath the name, a tagline 'Secure cryptocurrency transactions with messaging' is displayed in a smaller, light blue font, preceded by a small star icon. The main content area is a dark grey rounded rectangle. Inside, the word 'Welcome' is centered in a large white font. Below it, the text 'Access your secure crypto payment platform' is centered in a smaller white font. There are two buttons: a black 'Login' button and a purple-to-pink gradient 'Register' button. Below these are two input fields. The first is labeled 'Email' and contains an envelope icon and the text 'Enter your email'. The second is labeled 'Password' and contains a lock icon and the text 'Create a password'. Below the password field is a large purple-to-pink gradient button labeled 'Create Account'. At the bottom, there is a horizontal line with the text 'or sign in with Civic' in the center. Below this line is a large blue-to-purple gradient button labeled 'Sign In with Civic'.



CryptoSafeSend

✦ Secure cryptocurrency transactions with messaging

Welcome

Access your secure crypto payment platform

Login Register

Email

✉ Enter your email

Password

🔒 Create a password

Create Account

or sign in with Civic

Sign In with Civic

How to cite this article:

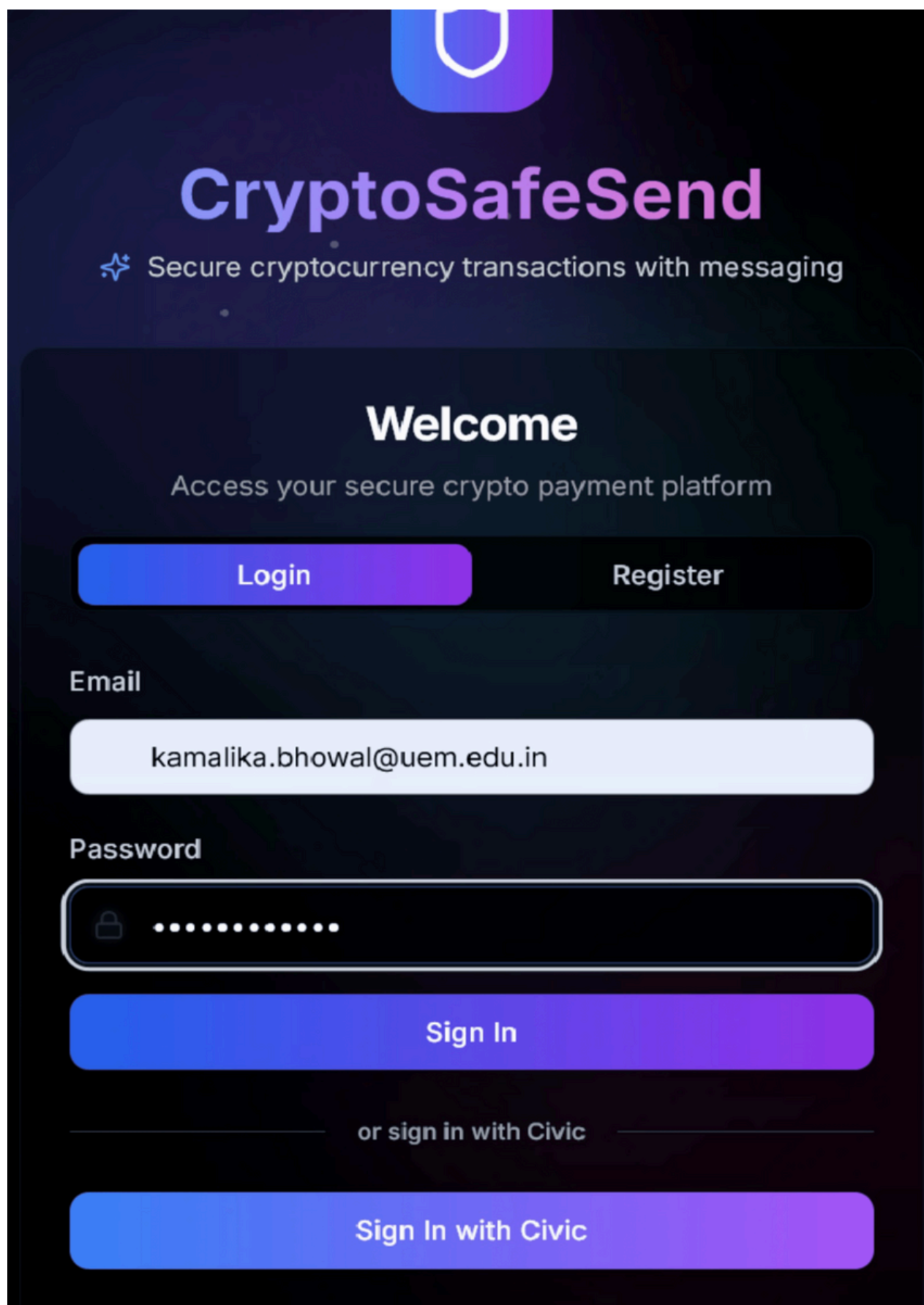
Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6> Page 7 of 37

FIGURE 3: User Registration Form.

Source: Screenshot from the CryptoSafeSend application (developed by the authors).

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6> Page 8 of 37



The image shows a digital interface for 'CryptoSafeSend'. At the top, there is a logo consisting of a blue and purple shield-like shape. Below the logo, the text 'CryptoSafeSend' is displayed in a large, bold, purple font. Underneath the name, a tagline reads 'Secure cryptocurrency transactions with messaging' in a smaller, light blue font, preceded by a small star icon. The main content area is a dark blue rounded rectangle. Inside, the word 'Welcome' is centered in a large white font. Below it, the text 'Access your secure crypto payment platform' is centered in a smaller white font. There are two buttons: 'Login' in a blue-to-purple gradient and 'Register' in a dark blue gradient. Below these buttons, there are input fields for 'Email' and 'Password'. The email field contains the text 'kamalika.bhowal@uem.edu.in'. The password field is masked with dots and has a lock icon on the left. Below the password field is a large 'Sign In' button in the same blue-to-purple gradient. At the bottom, there is a link 'or sign in with Civic' and a corresponding 'Sign In with Civic' button in the blue-to-purple gradient.

CryptoSafeSend

Secure cryptocurrency transactions with messaging

Welcome

Access your secure crypto payment platform

Login Register

Email

kamalika.bhowal@uem.edu.in

Password

Sign In

or sign in with Civic

Sign In with Civic

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6> Page 9 of 37

FIGURE 4: Email/Password Login User Interface With Civic Option.

Source: Screenshot from the CryptoSafeSend application (developed by the authors).

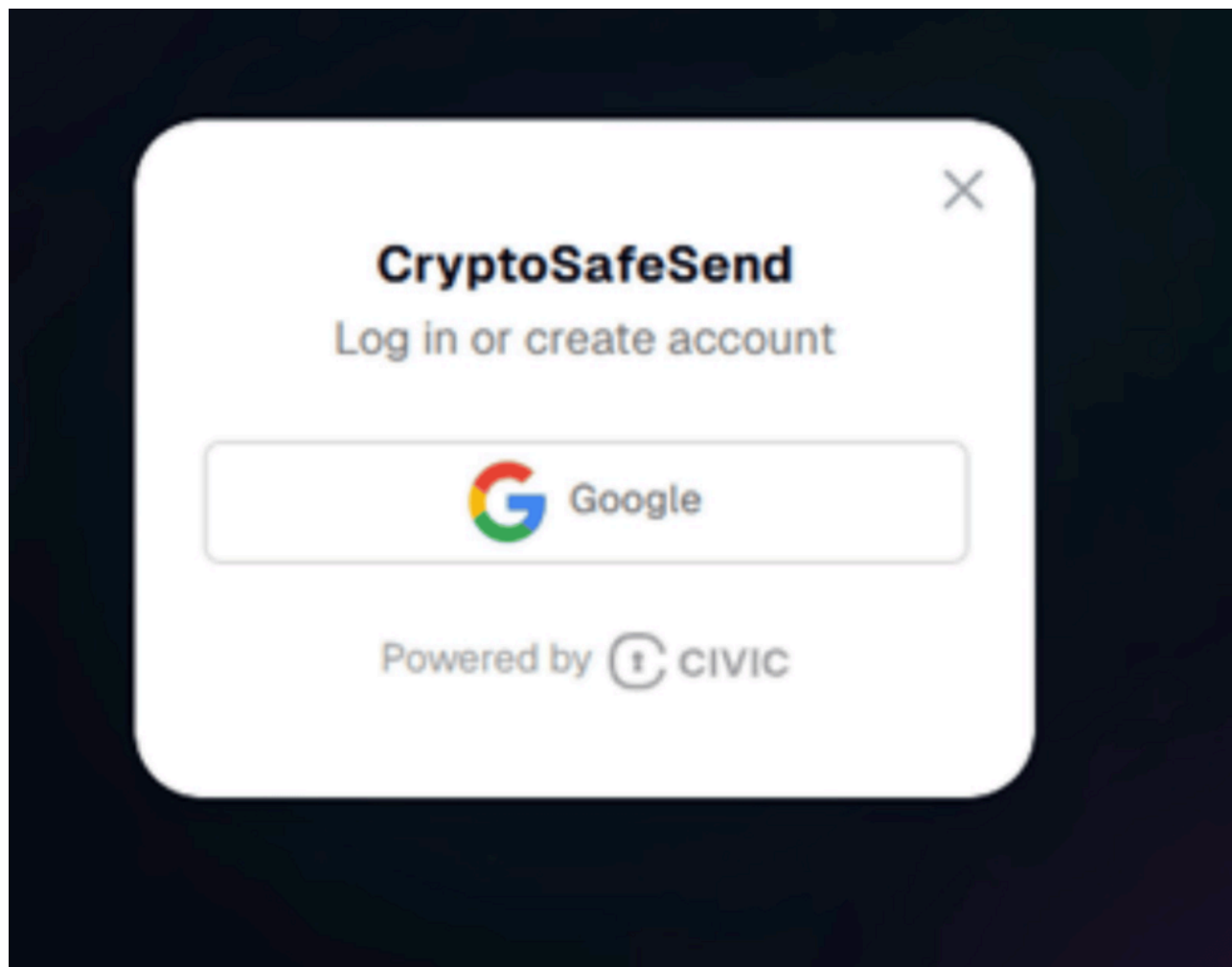


FIGURE 5: Google OAuth-Based Login Interface.

Source: Screenshot from the CryptoSafeSend application (developed by the authors).

Wallet integration and passcode setup

Following authentication, users are required to connect their MetaMask wallet to the application in order to initiate and authorize fund transfers, as illustrated in Figure 6. This integration enables secure communication between the application and the Ethereum blockchain without exposing or storing the user's private keys. MetaMask serves as a client-side wallet interface that signs transactions locally and broadcasts them to the blockchain network. For development and testing purposes, the system is configured to interact with a locally deployed Ethereum test network using Ganache. Ganache provides a personal blockchain environment that simulates smart contract deployment and transaction execution in a controlled setting. The local network is typically accessible through the RPC endpoint at <http://127.0.0.1:7545>. To establish connectivity, a custom network is added to MetaMask by specifying the Ganache RPC URL and the corresponding chain ID, commonly 1337 or 5777. Developers may then import a test account from Ganache into MetaMask using the provided private key for experimental purposes. This configuration facilitates direct interaction

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. *Cureus J Comput Sci* 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

between MetaMask and the local blockchain, enabling secure testing of transaction workflows and smart contract functionality prior to deployment on a public Ethereum network. Such an approach ensures reliability, minimizes deployment risks, and supports iterative development in a secure and controlled environment. Four essential inputs must be supplied by the sender in order to start a secure fund transfer, as shown in Figure 7: the recipient's Ethereum wallet address, the quantity of Ether to be transferred, a secret passcode, and the purpose of the transfer. To maintain its confidentiality, this passcode is hashed off-chain using a cryptographic hash function, Keccak-256. The CryptoSafeSend smart contract on the Ethereum network is then notified of the hashed passcode, transfer amount, and recipient details. By essentially creating a conditional escrow, this procedure makes sure that only the designated beneficiary can access the assets; the money stays locked in the contract until the recipient gives the right secret passcode. This method prevents unauthorized or misdirected transfers by combining off-chain passcode verification with blockchain immutability.

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. *Cureus J Comput Sci* 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

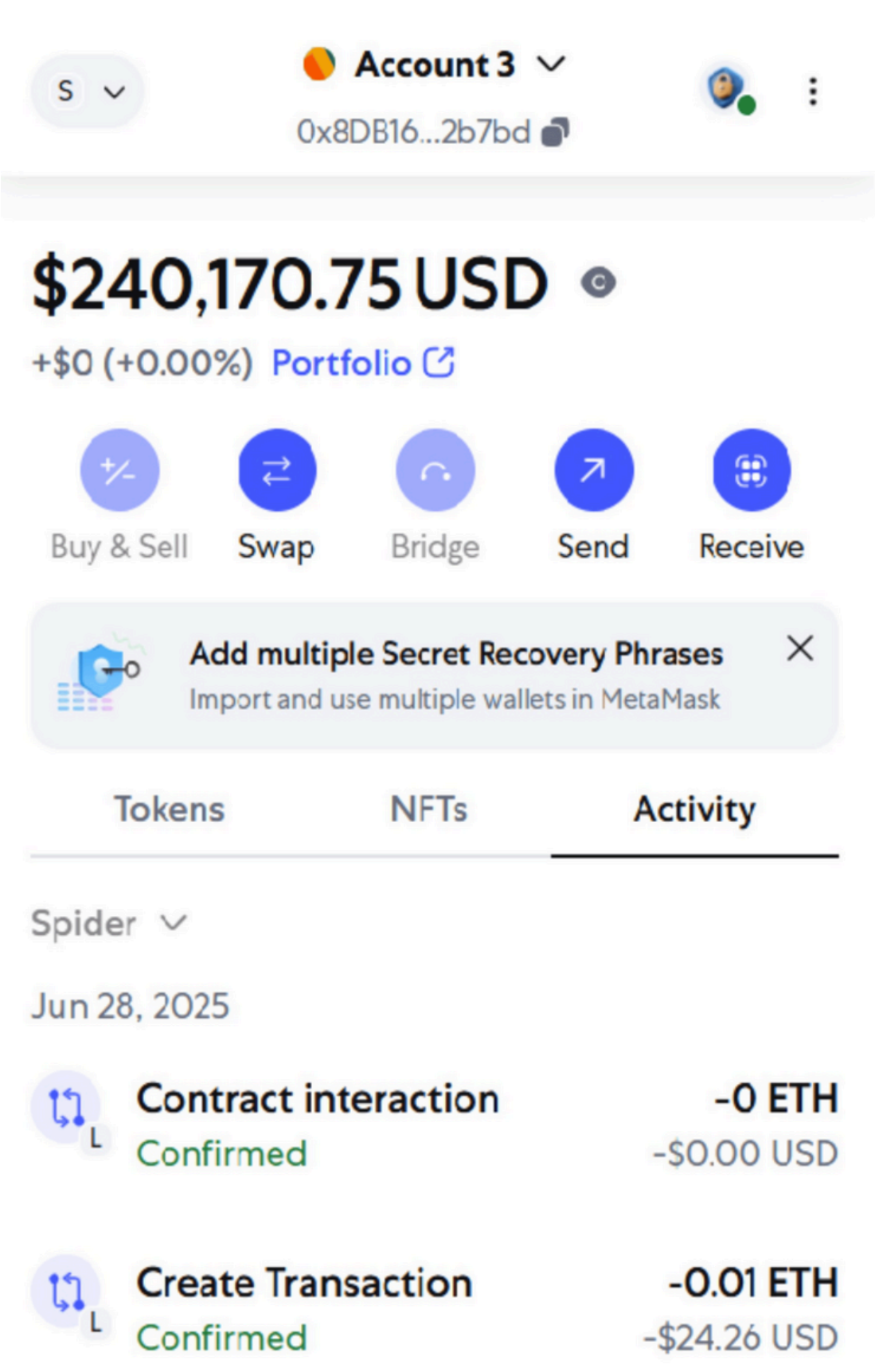


FIGURE 6: MetaMask Connection.

Source: Screenshot from the CryptoSafeSend application (developed by the authors).

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

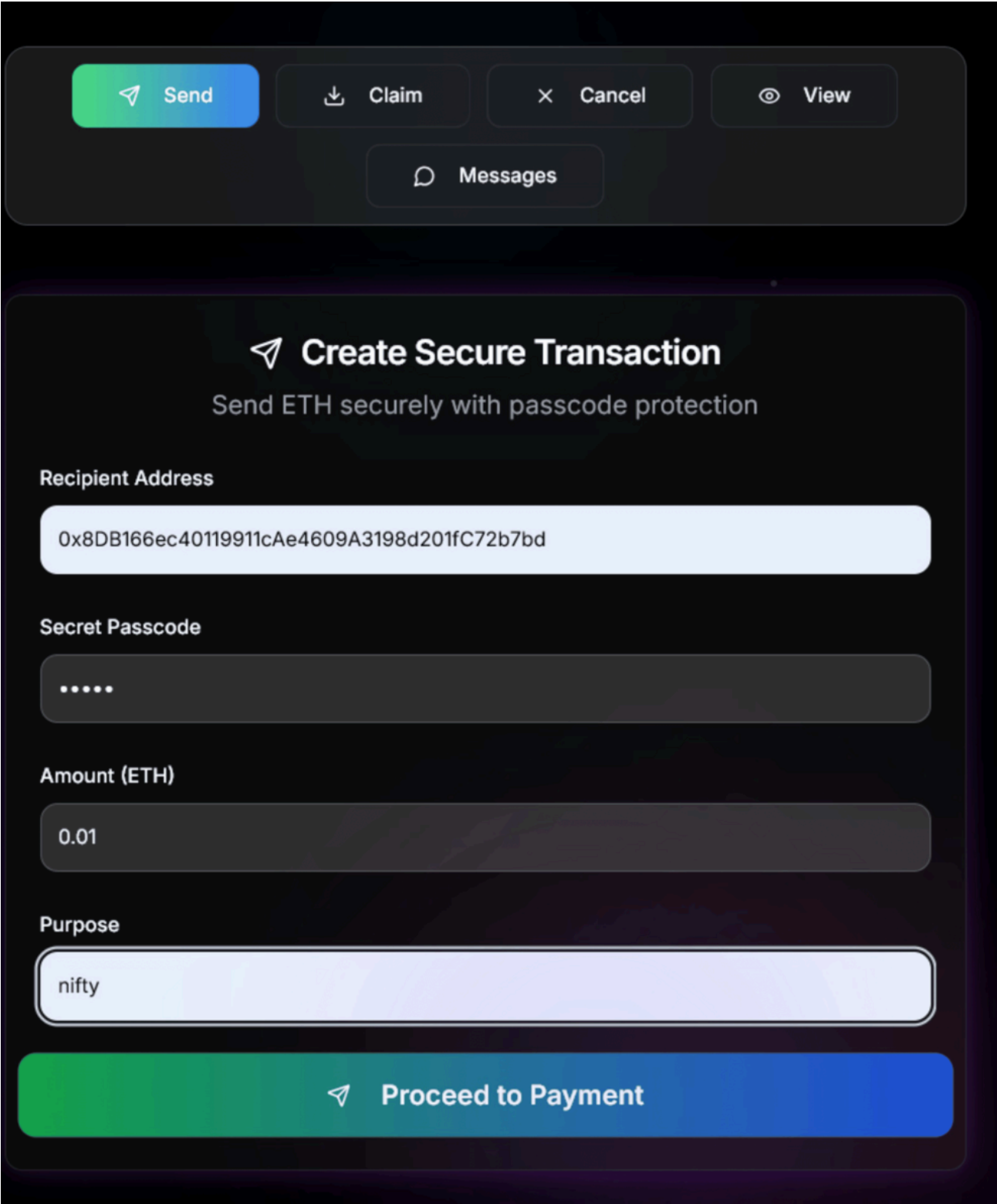


FIGURE 7: Create a Secure Transaction.

Source: Screenshot from the CryptoSafeSend application (developed by the authors).

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

Before starting a payment, users are presented with a clear and user-friendly summary of the transaction by the CryptoSafeSend transaction interface. This summary includes key details such as the transaction amount, recipient address, and escrow secret passcode. To further enhance security awareness, the system incorporates a voice warning feature that prompts users to listen to an audio alert before proceeding. This serves as an additional safeguard by reminding users to securely share the secret passcode and verify the recipient information, helping to prevent unauthorized transactions, as illustrated in Figure 8.

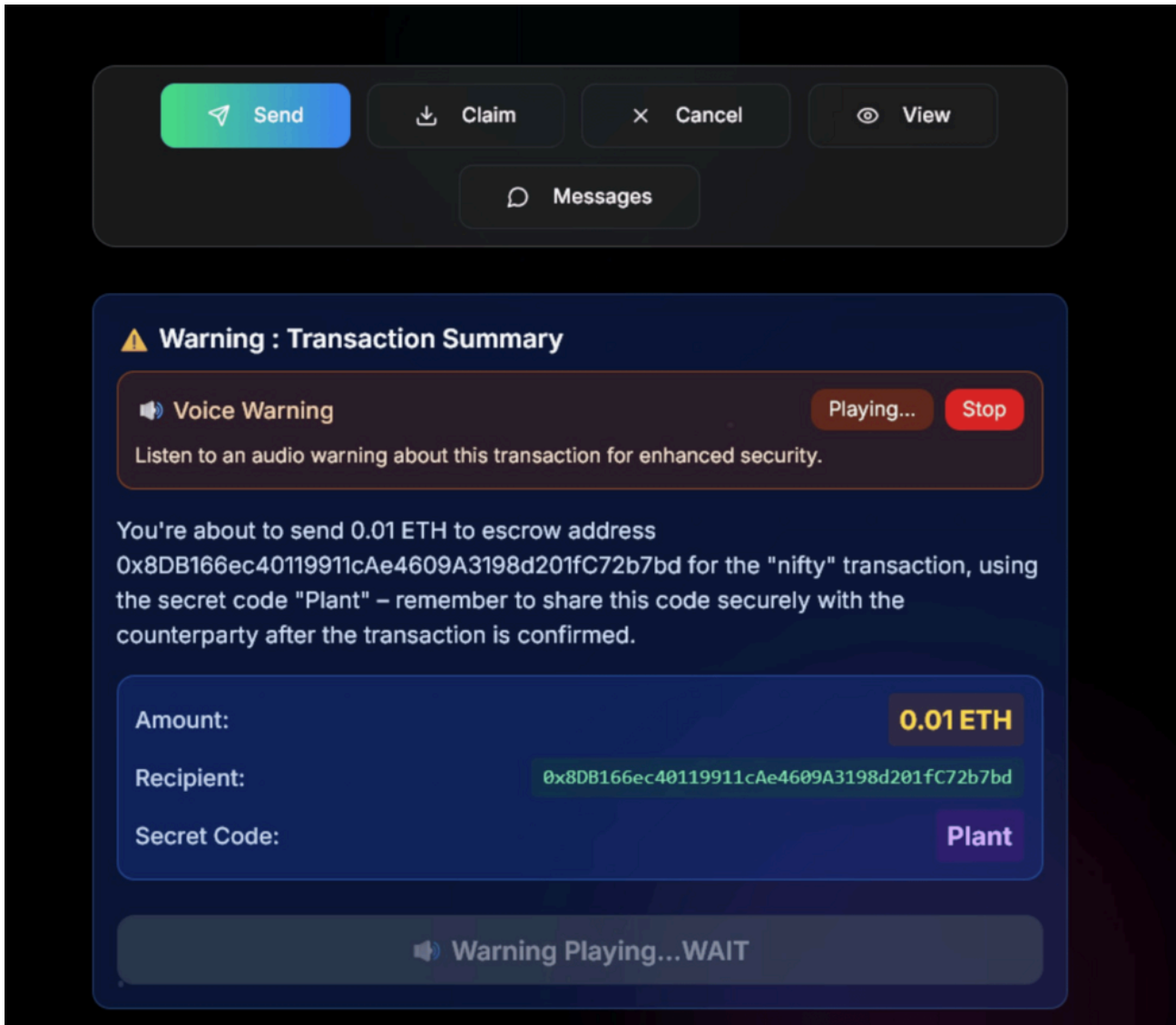


FIGURE 8: Security Warning Before Confirmation of Transactions.

Source: Screenshot from the CryptoSafeSend application (developed by the authors).

Off-chain messaging and passcode delivery

In the given example, the user intends to send 0.01 ETH to a specified Ethereum address, using the secret passcode "Plant", which was provided by the user during the transaction initiation process (refer to Figure 8). After clicking the "Proceed to Payment" button, the MetaMask wallet presents a transaction request screen, as shown in Figure 9, displaying details such as the amount of Ether being transferred and the associated network fee. This two-step verification process-comprising the on-screen summary followed by MetaMask's confirmation-ensures users retain full

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

control and transparency, thus minimizing the risk of accidental or malicious transfers. Thus, a transaction is created successfully, as shown in Figure 10. To securely process the transfer, users first connect their MetaMask wallet to the application. Then, the sender provides the recipient's Ethereum address, the amount of Ether, and the secret passcode. The passcode is not stored directly; instead, it undergoes a secure process, as shown in Figure 11.

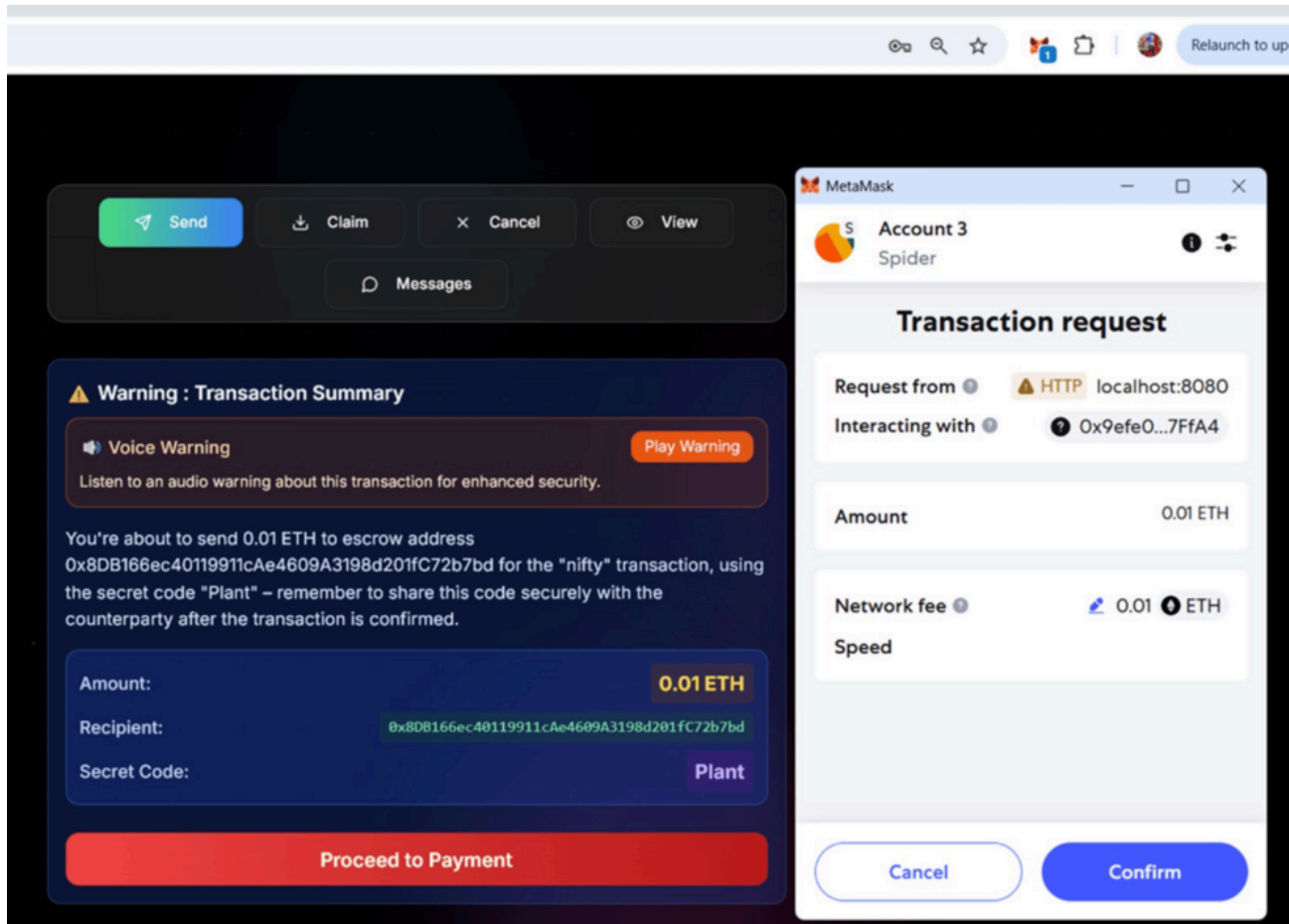


FIGURE 9: Transaction Request for Confirmation.

Source: Screenshot from the CryptoSafeSend application (developed by the authors).

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

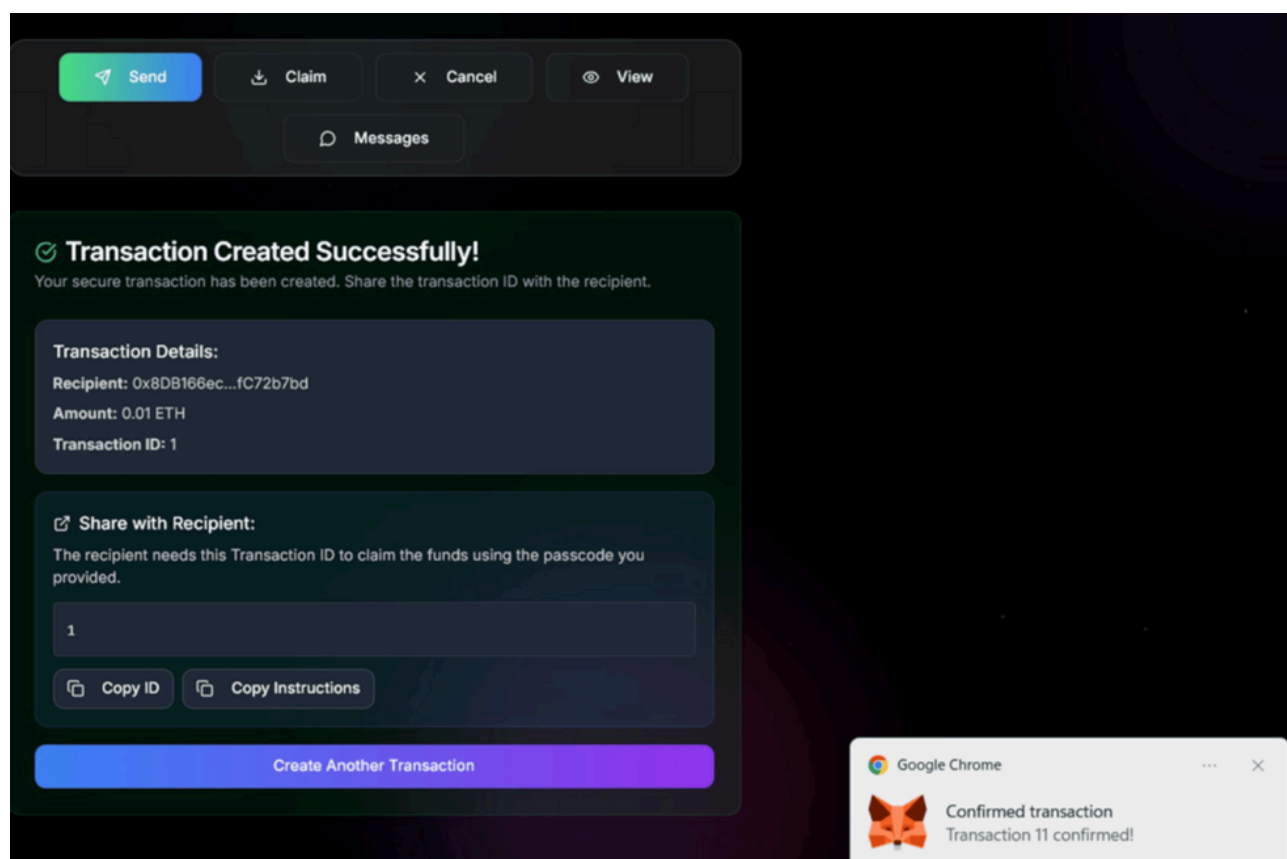


FIGURE 10: Confirmation of Transaction.

Source: Screenshot from the CryptoSafeSend application (developed by the authors).

A multi-layered cryptographic technique is used in CryptoSafeSend's passcode protection mechanism to guarantee both verifiability and confidentiality throughout the transaction process. First, a Password-Based Key Derivation Function (PBKDF2) is used to securely transform the user-defined passcode. This procedure creates a strong, user-specific cryptographic key by using the user's Firebase Unique Identifier (UID) as a cryptographic salt. By adding computational complexity during key derivation, PBKDF2 improves resistance against rainbow table and brute-force attacks. After key derivation, the generated cryptographic key is used to encrypt the passcode on the client side using the Advanced Encryption Standard operating in Galois/Counter Mode (AES-GCM). The browser-native Web Crypto API is used to implement this encryption process, guaranteeing that the plaintext passcode is never revealed during storage or transmission. By combining encryption with authenticated encryption techniques, AES-GCM offers data integrity and confidentiality. The encrypted passcode and its corresponding Initialization Vector (IV) are encoded using Base64 representation to enable safe transmission and storage. As part of the off-chain data management system, these encoded values are then kept in the Firebase Firestore database. Sensitive user credentials are safeguarded even in the event that unauthorized parties gain access to the database by storing encrypted values instead of plaintext data. For decentralized verification within the blockchain environment, only the Keccak-256 cryptographic hash of the passcode is stored on-chain within the smart contract. During the claim process, the recipient provides the plaintext passcode, which is hashed by the smart contract and compared against the stored hash value. Because the blockchain stores only the hash representation, the original passcode is never revealed on-chain. This architecture combines off-chain encryption with on-chain hash verification, thereby strengthening resistance to brute-force attacks while preserving the confidentiality and integrity of sensitive authentication information.

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

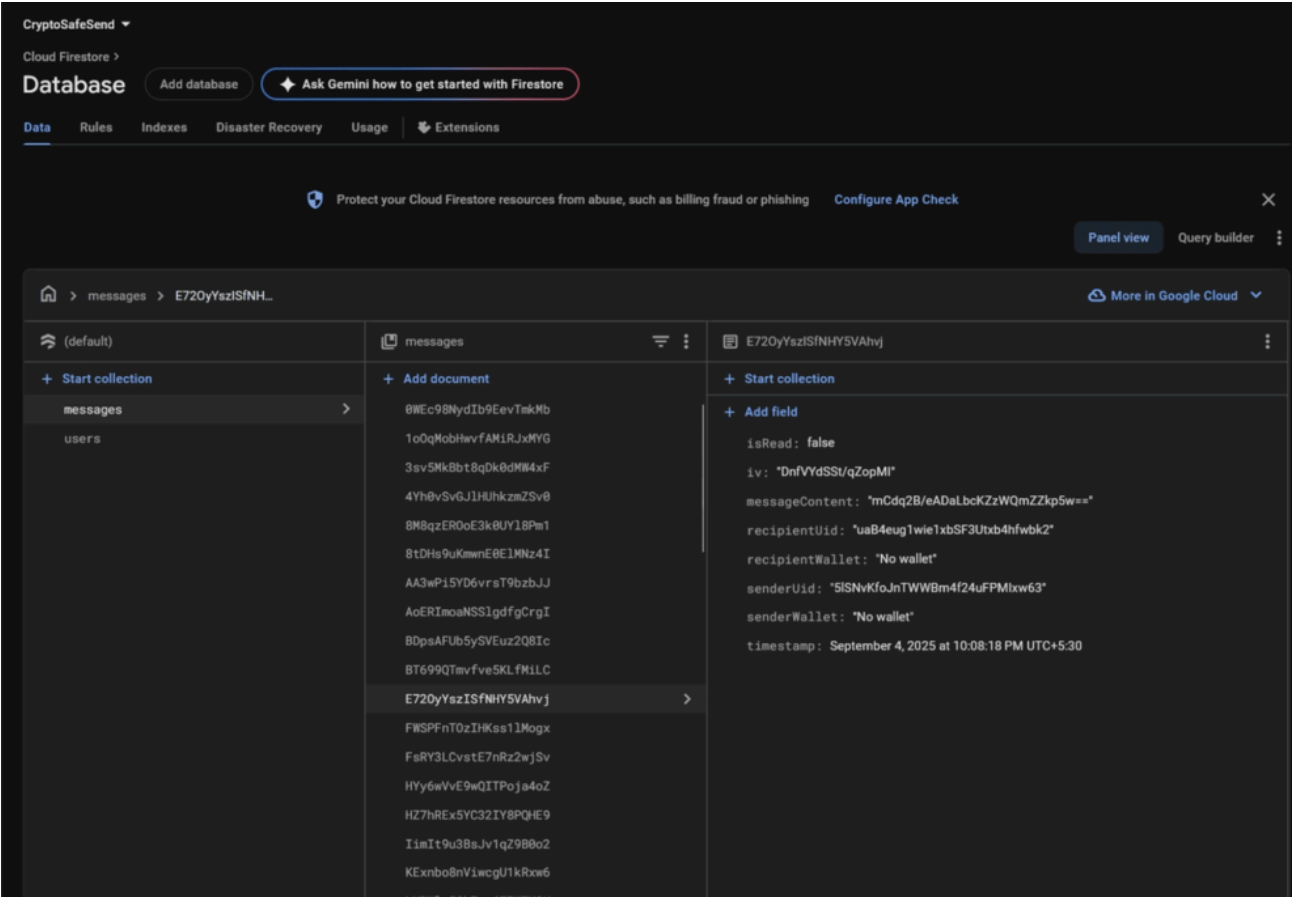
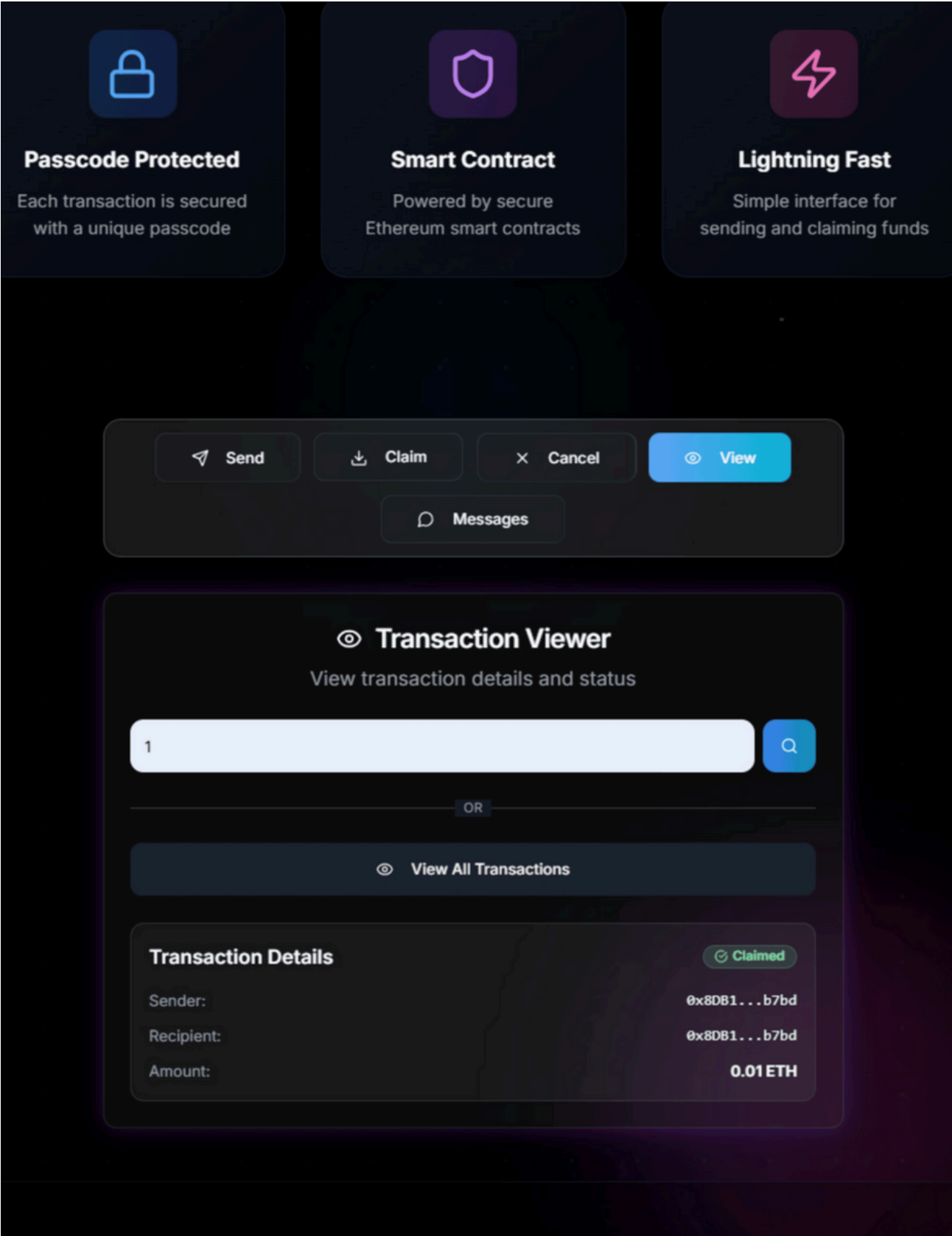


FIGURE 11: Security of Transaction Interface.

Source: Author’s own work.

Users can view all initiated escrow transactions, as shown in Figure 12, in real time and transparently through CryptoSafeSend's Transaction Viewer interface. Users have the option to view a list of all recent transactions or search for a specific transaction using its unique Transaction ID, as shown in the image. Important details like the transaction amount, sender and recipient addresses, and the current status - such as "Pending" - are displayed for each entry.

How to cite this article:



How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

FIGURE 12: Transaction Viewer.

Source: Screenshot from the CryptoSafeSend application (developed by the authors).

By allowing users to either claim or cancel transactions in the escrow process, CryptoSafeSend guarantees flexibility and security. By entering the relevant Transaction ID, a sender can cancel a pending transaction, as demonstrated in the Cancel Transaction interface (Figure 13). The smart contract returns the locked funds to the sender's wallet after confirmation. When a passcode sharing error occurs or the recipient fails to claim the funds within the anticipated timeframe, this feature is especially helpful. In addition to providing an additional degree of user control, the cancellation mechanism guards against assets being permanently lost as a result of errors or inactivity.

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

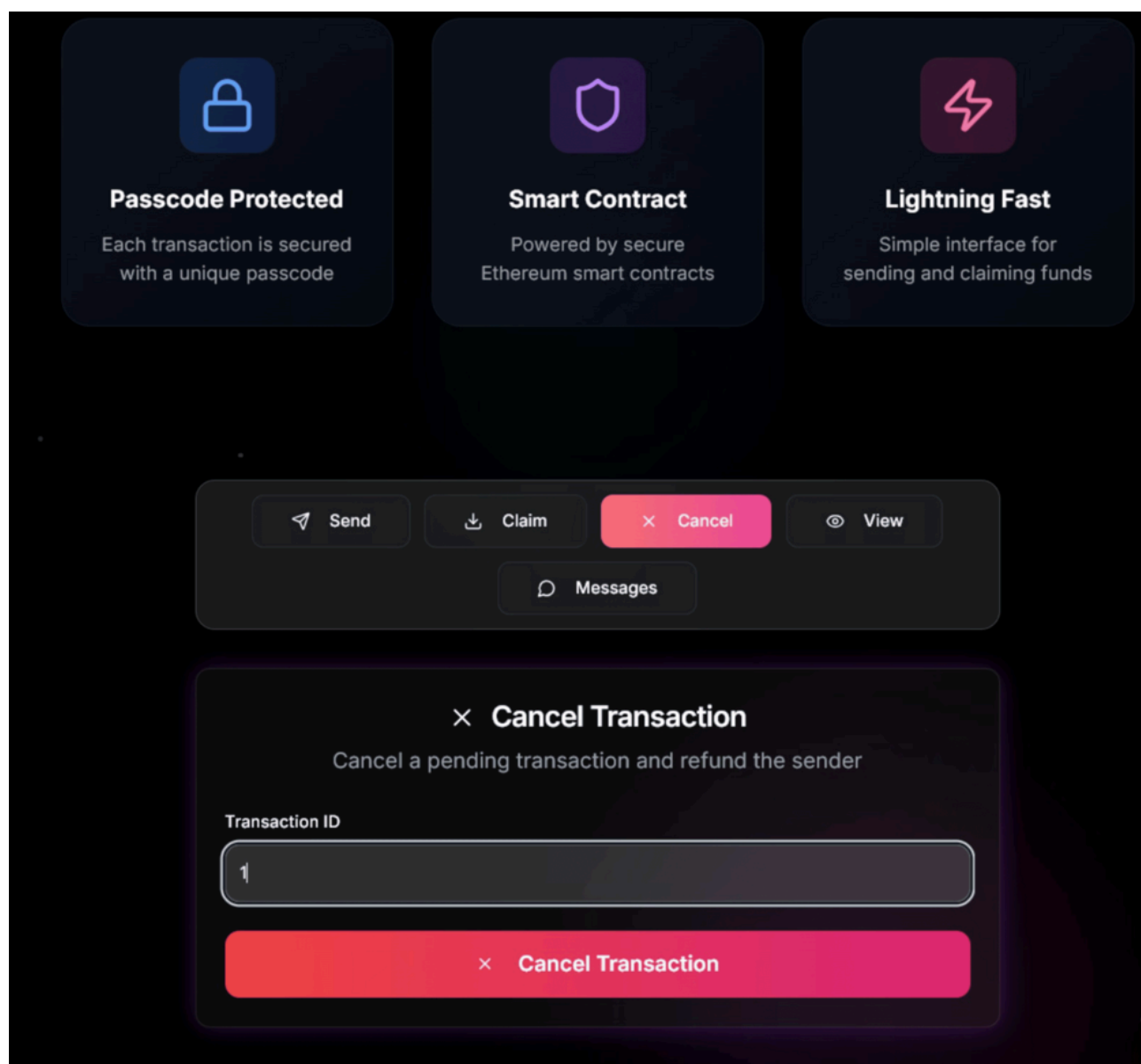


FIGURE 13: Cancel a Transaction.

Source: Screenshot from the CryptoSafeSend application (developed by the authors).

On the other hand, the Claim Funds interface depicted in Figure 14 allows recipients to safely access escrowed funds. The recipient needs to enter the correct Transaction ID and the related secret passcode that was shared off-chain by the sender in order to successfully claim a transaction. After that, the smart contract confirms the input by comparing the supplied passcode with the hashed one that is kept on-chain. The money is automatically moved to the recipient's wallet if the match is successful. By combining an off-chain secret with an on-chain hash, this dual-authentication method makes sure that only the designated recipient can access the money, avoiding fraudulent claims and strengthening the CryptoSafeSend platform's security model.

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

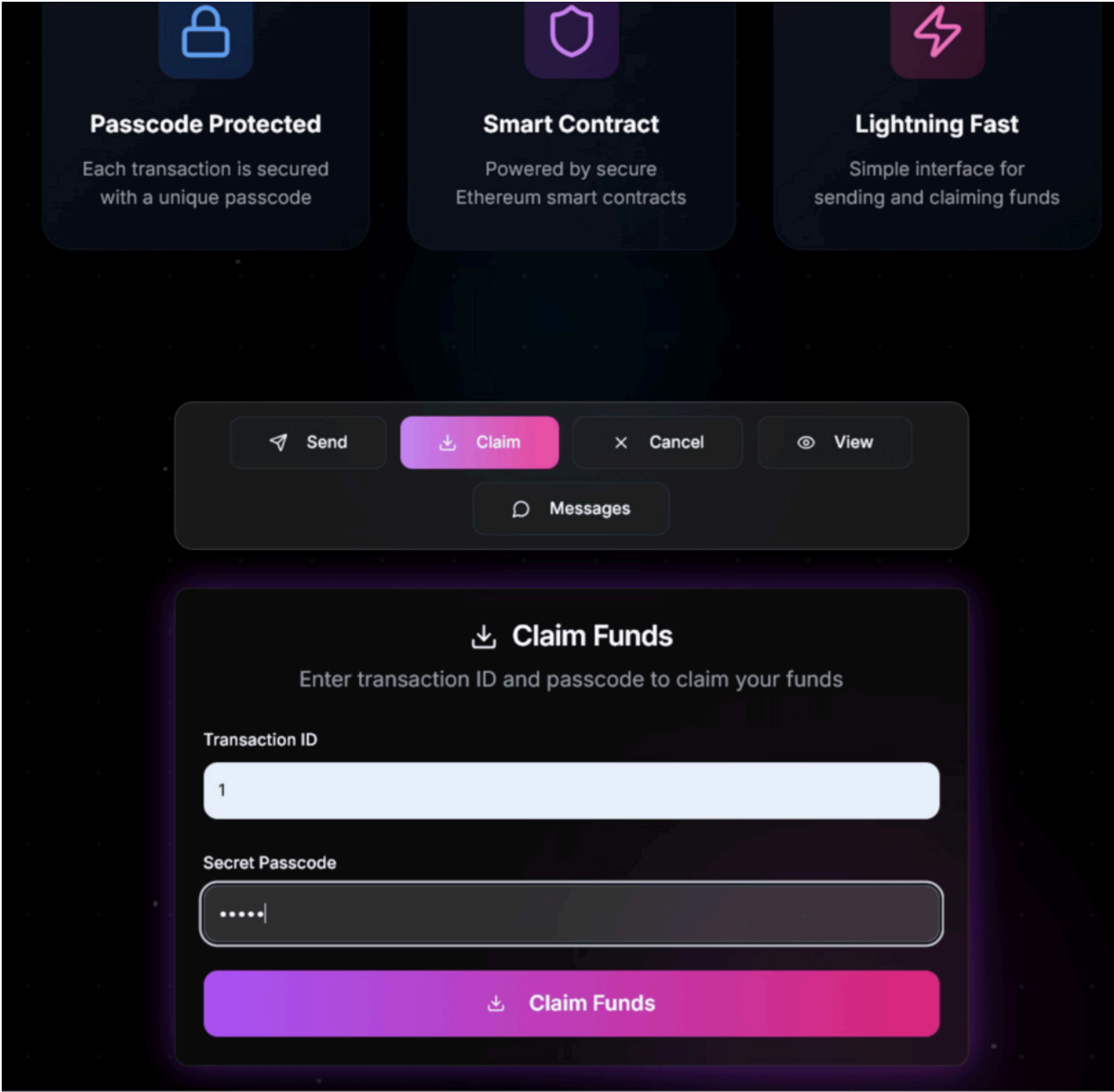


FIGURE 14: Claiming of Funds.

Source: Screenshot from the CryptoSafeSend application (developed by the authors).

Security and testing

CryptoSafeSend incorporates multiple security layers to ensure the integrity, confidentiality, and correctness of its escrow-based transaction system. Frontend validation of recipient Ethereum addresses is enforced using industry-standard cryptographic libraries such as *ethers.js*, significantly reducing the risk of human input errors. Upon transaction initiation, the sender’s wallet address is immutably recorded in the smart contract state, and strict access control ensures that only the original sender can cancel a transaction, as enforced by sender verification at the contract level as shown in the below code snippet:

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

```
\\(require( msg.sender == transaction.sender, "Only the sender can cancel the transaction." );\\)
```

Unauthorized cancellation attempts are automatically reverted, preserving escrow integrity. Fund claims are permitted only after successful passcode verification, and all state transitions are strictly enforced to prevent fund loss or indefinite locking. Contract correctness is validated through unit testing in the Remix IDE, while gas efficiency is achieved by minimizing storage operations and avoiding expensive looping constructs.

To further protect off-chain passcodes, CryptoSafeSend employs browser-native symmetric encryption using the Web Crypto API. Encryption keys are derived via PBKDF2 with the user's Firebase UID as a unique salt, ensuring user-specific key isolation and resistance to brute-force attacks. AES-GCM is used for authenticated encryption and decryption within the browser, and the resulting ciphertext and initialization vector are Base64 encoded for secure storage and transmission via Firebase Firestore. These cryptographic operations are encapsulated in a dedicated module, promoting modularity and maintainability. Collectively, this architecture, as shown in Figure 15, provides strong guarantees of security, efficiency, and reliability without compromising usability or performance.

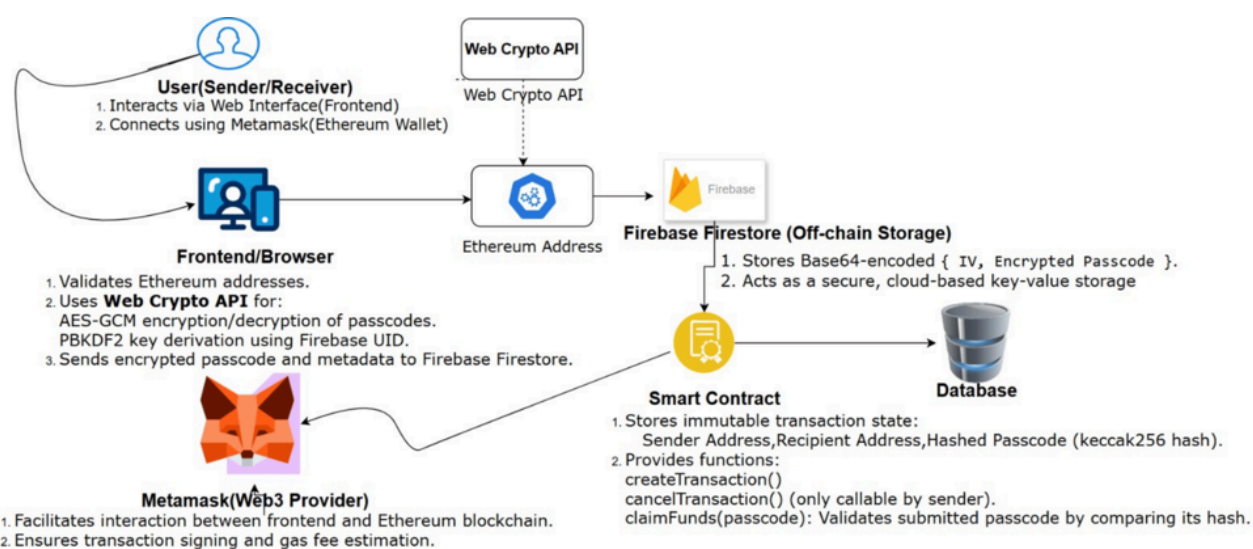


FIGURE 15: Security and Testing Architecture.

Source: Author's own work.

Testing and validation

A series of contract interactions started by the same sender address can also be seen in Ganache's Transactions tab in Figure 16. A combination of fund transfers and administrative actions (such as establishing or terminating escrow with a value of 0 ETH) are included in the transactions that are shown. It is noteworthy that certain transactions display a value of 1 ETH (in wei), signifying that the smart contract has actually locked funds. One transaction used 121,745 gas units, for example, demonstrating that the operation involved sophisticated logic like hash verification and writing to the blockchain, which are typical in secure escrow systems. These different gas usages across calls verify that smart contract methods are invoked correctly and show the relative computational cost of each function.

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

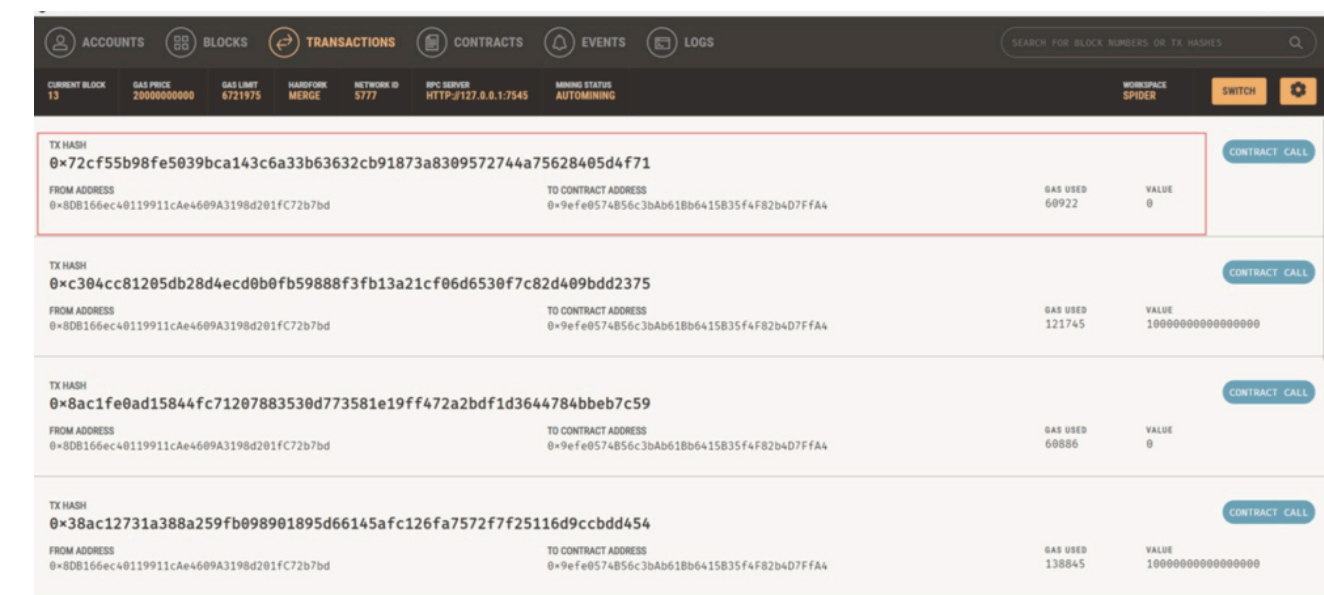


FIGURE 16: Ganache Transactions.
Source: Screenshot captured during testing using Ganache.

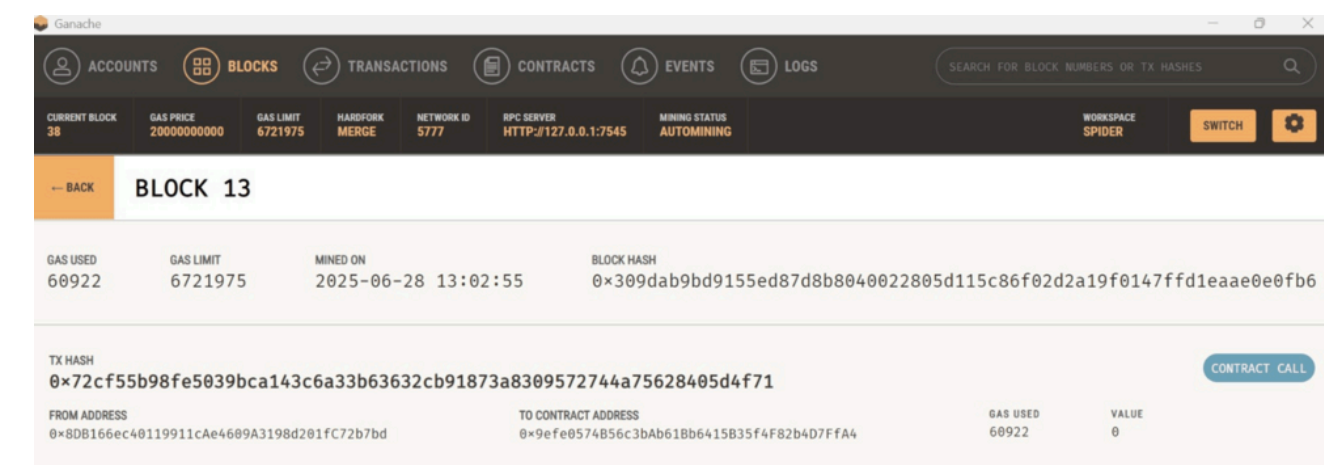


FIGURE 17: Blocks Mined
Source: Screenshot captured during testing using Ganache.

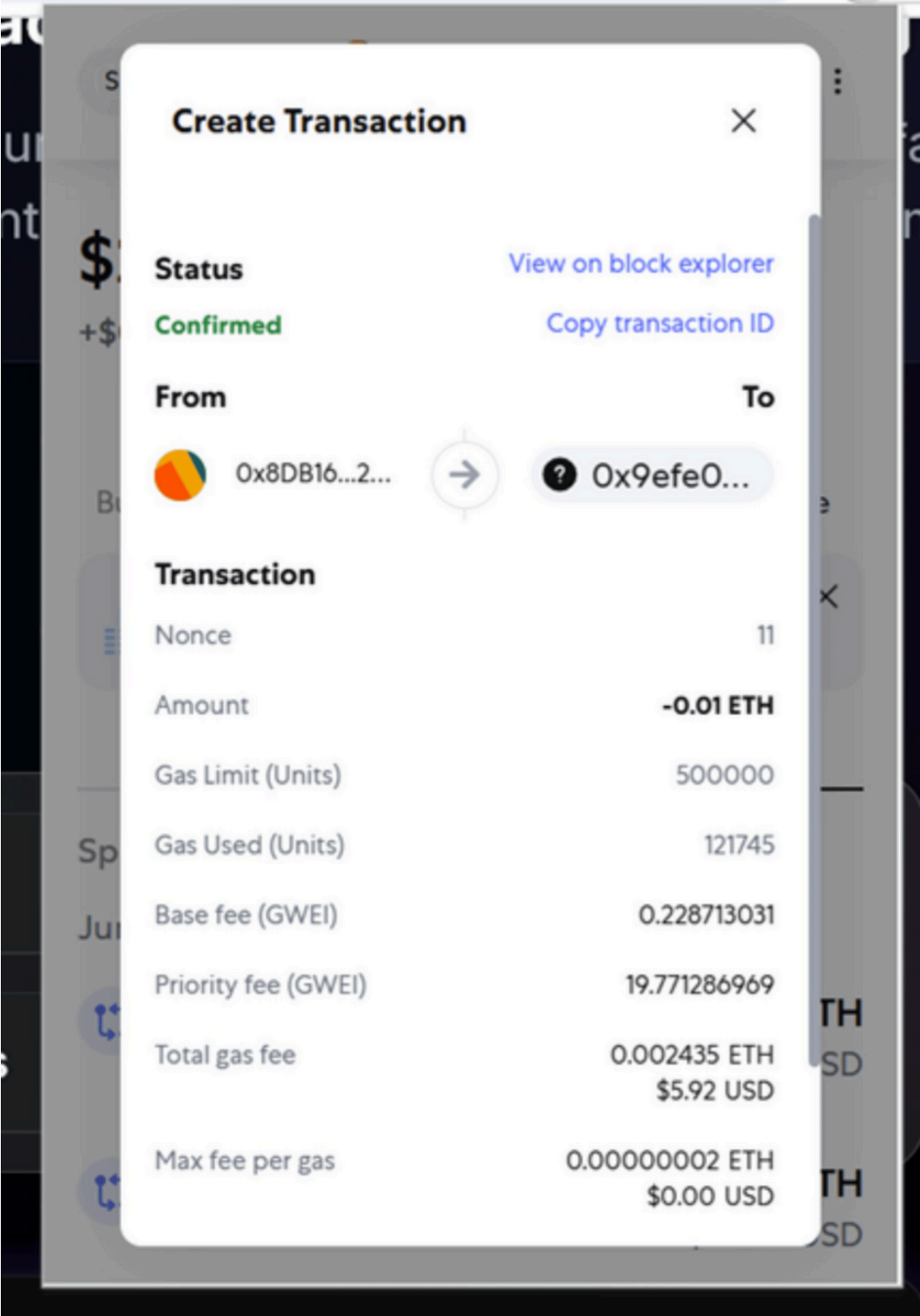
Ganache's local blockchain simulation is used to verify that CryptoSafeSend's transaction logic is being executed correctly. A transaction from the sender's wallet address (0x8D8166...) to the deployed smart contract address (0x9efe0...) is shown in Block 13 of the Ganache Block view. The block contains key blockchain information like the exact timestamp of mining, the block hash, and the amount of gas used (60,922 units) (Figure 17). This low-level view demonstrates that the conditional escrow logic was carried out and stored immutably within the Ethereum test network environment, confirming the transaction's successful inclusion in the chain.

A live user-initiated transaction is also recorded by the MetaMask confirmation panel, as shown in Figure 18, confirming that 0.01 ETH was transferred from the sender wallet to the CryptoSafeSend smart contract. The transaction metadata contains the gas limit, the amount of gas used (121,745 units), and the breakdown of fees in both GWEI and USD. The

status is marked as "Confirmed." This last confirmation completes the full trace from frontend initiation to blockchain validation, ensuring that the transaction was successful and matching the data displayed in Ganache. The CryptoSafeSend platform's dependability and transparency are reinforced by the smooth integration of smart contract execution, wallet authorization, and blockchain confirmation. Some Ganache-performed transactions for the CryptoSafeSend applications are shown in Table 7.

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. *Cureus J Comput Sci* 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>



How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

FIGURE 18: Corresponding Metamask Transactions.

Source: Screenshot from the Metamask Wallet connected to CryptoSafeSend application (developed by the authors).

Tx ID	Block No.	ETH Value	Gas Used	Status
Tx-01	12	0 ETH	60,922	Confirmed
Tx-02	13	1 ETH	121,745	Confirmed
Tx-03	14	0 ETH	118,304	Confirmed

TABLE 1: Sample Ganache Transaction Validation Results.

End-to-end workflow

The CryptoSafeSend Secure Escrow Transaction Algorithm as illustrated in Figure 19 explains how the CryptoSafeSend platform's secure transaction workflow works. It shows the order of operations, which include wallet authentication, PBKDF2 and AES-GCM passcode encryption, secure escrow creation through the smart contract, and the claim verification mechanism that uses Keccak-256 hash matching.

The CryptoSafeSend end-to-end workflow starts with user authentication via Google OAuth or email/password login, as illustrated in Figure 20. This is followed by a secure wallet connection via MetaMask. By providing the recipient's wallet address, the transfer amount, and a secret passcode, the sender starts an escrow transaction. The recipient receives the plaintext passcode via an off-chain channel, but its cryptographic hash is kept on-chain. The recipient submits the passcode and the transaction identifier to claim the funds after receiving them. The smart contract releases the escrowed Ether to the recipient following a successful hash verification. A local Ethereum simulator like Ganache can be used to independently verify each transaction event, which is immutably recorded on the blockchain.

How to cite this article:

Algorithm: CryptoSafeSend Secure Escrow Transaction

Input:

U – Authenticated user
R – Recipient Ethereum address
A – Ether amount
P – Secret passcode
UID – Firebase user identifier

Output:

Secure escrow-based cryptocurrency transaction

Procedure:

Transaction Initialization Phase

- (1) Connect the user wallet through MetaMask.
- (2) Verify the recipient wallet address R.
- (3) If the address is invalid, terminate the transaction.
- (4) Generate cryptographic key K using PBKDF2 with passcode P and salt UID.
- (5) Encrypt passcode using AES-GCM to produce ciphertext C and IV.
- (6) Encode C and IV using Base64.
- (7) Store encrypted passcode and IV in Firebase Firestore.
- (8) Compute passcode hash H using Keccak-256.
- (9) Call smart contract function createTransaction(R, H, A).

Transaction Management Phase

- (10) If sender cancels the transaction, verify sender identity.
- (11) Execute cancelTransaction() to refund funds.

Claim Verification Phase

- (12) Recipient submits passcode during claim.
- (13) Recalculate hash H'.
- (14) If $H' = H$ then release escrowed funds.
- (15) Else reject claim request.

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

FIGURE 19: CryptoSafeSend Secure Transaction Workflow Algorithm.

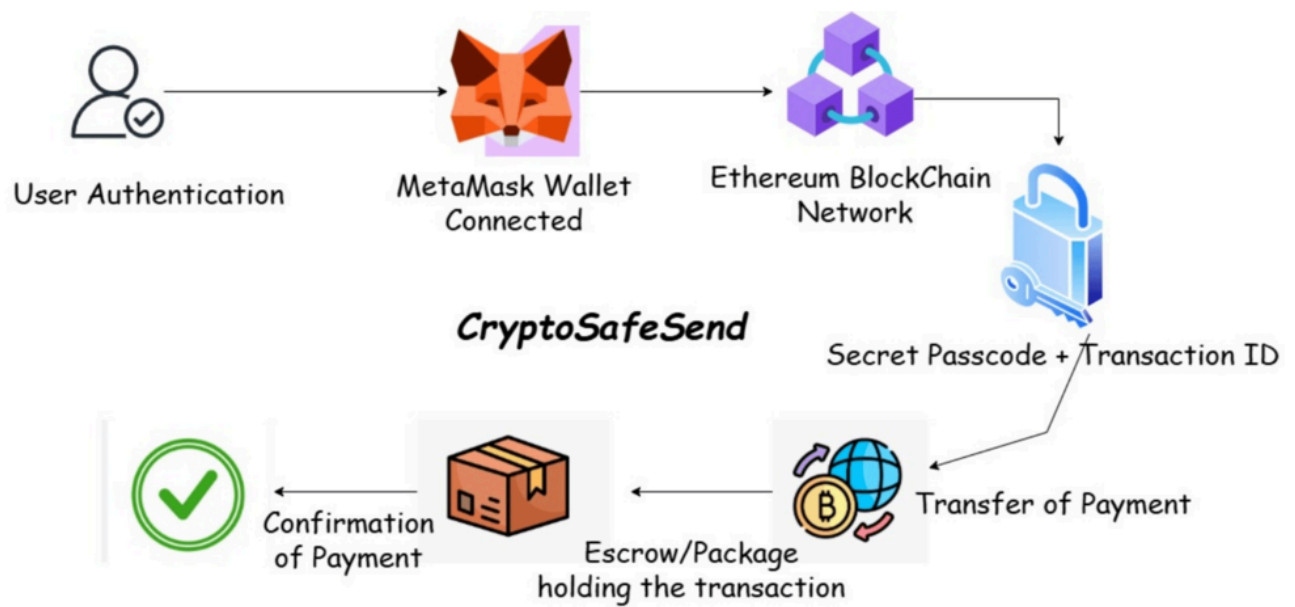


FIGURE 20: End to End Workflow.

Source: Author's own work.

Results And Discussion

In this section, we have discussed our experimental results, details of transactions, their execution times, and relative transaction costs. We have analyzed the performance of our CryptoSafeSend app based on the Google PageSpeed score, as shown in Figure 21. We have also analyzed similar existing platforms and their performance based on PageSpeed Insights by Google, as illustrated in Figure 22.

Experimental setup

In order to evaluate CryptoSafeSend's performance, security, and cost-effectiveness in a real-world decentralized finance setting, an experimental study was carried out. The blockchain network, the off-chain secure messaging system, and the client-side performance environment comprised the three main levels of the setup.

The environment of smart contracts and blockchain: The SafeSend smart contract, written in Solidity (v0.8.18), was used to implement the fundamental mediator logic. The contract was compiled and optimized using the Remix IDE environment. Local unit testing was carried out on a Ganache-simulated blockchain for functional validation. The proposed CryptoSafeSend smart contract was deployed and evaluated on the Ethereum Sepolia Testnet (Contract Address: 0x7dd1cb4c97A9Bf8A24d040c6F44b316E093B12e3). The main Web3 wallet provider for gas management and transaction signing was MetaMask.

Software Development and Implementation: A contemporary tech stack consisting of TypeScript, Vite, and React was used to create the frontend application. The platform was implemented on Netlify, utilizing its worldwide Content Delivery Network and automated SSL management to guarantee high availability and low latency. Firebase Firestore was

How to cite this article:

used to manage off-chain data, including secure passcode exchange, and Firebase Authentication offered secure user identity management.

Cryptographic configuration and security: Cryptographic keys and security passcodes were never kept in plaintext to guarantee end-to-end confidentiality. The Web Crypto API was used in the experimental setup for client-side authenticated encryption (256-bit AES-GCM). Using 10,000 iterations of the Password-Based Key Derivation Function 2 (PBKDF2) and the user's distinct Firebase UID, encryption keys were generated. During the claim process, on-chain validation used the Keccak-256 cryptographic hash function to confirm passcodes.

Three main quantitative dimensions were used to thoroughly assess the CryptoSafeSend system's performance and technical feasibility. To determine the economic affordability and efficiency of the smart contract operations, gas complexity was first examined in terms of both absolute gas units and Gwei. Second, the temporal requirements for on-chain block confirmations were compared to the off-chain Firebase messaging layer's delivery speeds in order to evaluate network latency.

Lastly, the Google Lighthouse and PageSpeed Insights frameworks were used to benchmark the overall performance of the web interface, emphasizing Search Engine Optimization, adherence to contemporary web development best practices, and crucial rendering metrics like First Contentful Paint and Largest Contentful Paint. These metrics also offered a long-term perspective on the platform's stability and ability to manage safe, decentralized asset transfers in a variety of network scenarios.

Performance Analysis

The CryptoSafeSend application's performance evaluation in desktop mode, as shown in Figure 21, shows effective front-end rendering and robust optimization. The time it takes for the first visible content, like text or images, to appear on the screen after the page starts loading is known as the First Contentful Paint, and it was measured at 1.0 seconds. Improved perceived performance and quick initial rendering are indicated by this low value. The largest visible interface element loads quickly, allowing for timely user interaction, as evidenced by the Largest Contentful Paint, which was recorded at 1.3 seconds.

The main thread's script execution caused very little delay, as evidenced by the Total Blocking Time of 40 milliseconds, which guarantees high application responsiveness. Furthermore, the Speed Index was recorded at 1.7 seconds, indicating that the loading process renders the visible content gradually and smoothly. A consistent and user-friendly interface is enhanced by the Cumulative Layout Shift score of 0.026, which denotes high visual stability with few unexpected layout shifts.

With a Google Lighthouse Accessibility score of 98, the application demonstrated strong support for inclusive usability across a range of user groups. Additionally, the platform received a Best Practices score of 100, indicating compliance with contemporary web development standards, such as HTTPS implementation, secure coding techniques, and appropriate security headers. A Search Engine Optimization score of 100 further shows that the platform has the proper HTML structure and metadata to be fully optimized for search engine indexing.

The platform is deployed on Netlify, which provides global Content Delivery Network distribution, default SSL encryption, automated build pipelines, and edge-based delivery. This infrastructure ensures low latency, high availability, and reliable uptime, thereby supporting the secure and efficient operation of the CryptoSafeSend platform.

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

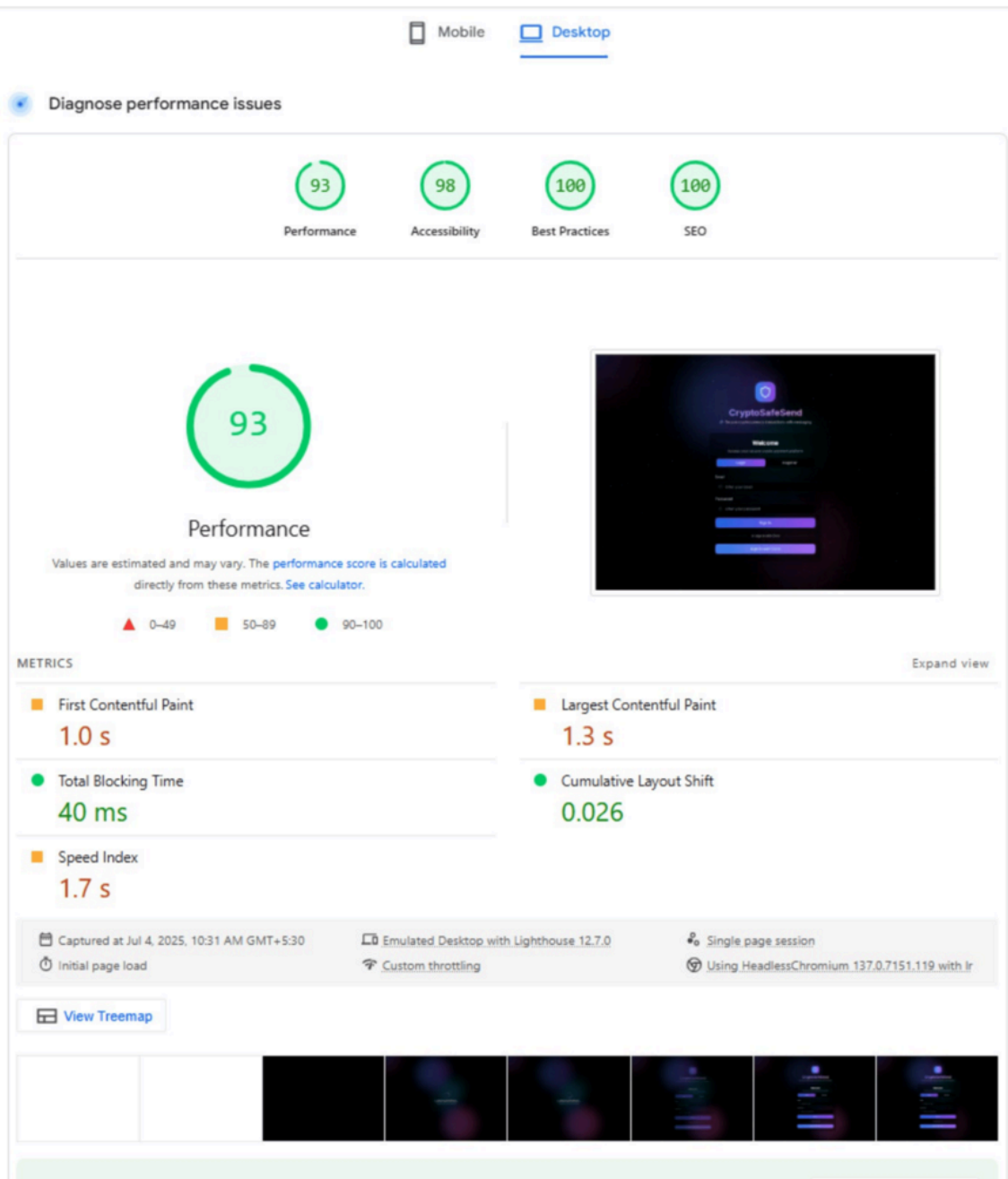


FIGURE 21: Performance Analysis of the CryptoSafeSend Interface Using Google PageSpeed Insights.

Source: Screenshot captured using Google PageSpeed Insights. Available at: <https://pagespeed.web.dev/>

Gas complexity and consumption of CryptoSafeSend

The gas consumption of *CryptoSafeSend* transactions was evaluated on the Ethereum test environment to assess cost efficiency. Gas usage varies with transaction type and contract complexity, typically ranging from several hundred thousand to just over one million gas units. The most gas-intensive operations involve on-chain storage of hashed passcodes and ether transfers, while conditional checks and state updates incur comparatively lower costs. Overall,

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. *Cureus J Comput Sci* 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

CryptoSafeSend achieves an effective balance between gas efficiency and security, making it suitable for secure, small-value escrow transactions in decentralized finance applications. The observed gas consumption characteristics are summarized in Table 2.

Transaction	ETH	Gas	Gwei	Status
Create Escrow	1	121,745	0.25	Confirmed
Claim Funds	0.01	98,312	0.24	Confirmed

TABLE 2: Gas Consumption of Representative CryptoSafeSend Transactions.

Comparison of secure crypto transfer platforms performance

Figure 22 presents a comparative evaluation of key security attributes across three escrow and transaction mediation approaches: CryptoSafeSend, Hashed Time Lock Contracts (HTLC), and custodial escrow systems. The vertical axis represents the level of support for each security attribute, where a value of 1.0 indicates full support, 0.5 indicates partial support, and 0.0 indicates no support.

Reentrancy protection is a critical security feature that prevents malicious contracts from repeatedly invoking vulnerable functions before the initial execution completes. Such attacks can result in unauthorized fund withdrawal or contract state manipulation. Both CryptoSafeSend and HTLC demonstrate full protection against reentrancy attacks by implementing secure contract execution logic and appropriate validation mechanisms. In contrast, custodial escrow systems rely on centralized intermediaries and do not inherently provide smart contract-level reentrancy safeguards.

Front-running resistance refers to the ability to prevent adversaries from exploiting pending transaction information to gain an unfair advantage by manipulating transaction ordering. CryptoSafeSend provides full resistance to front-running attacks by using secure passcode-based verification and controlled escrow release mechanisms. HTLC and custodial escrow systems demonstrate partial resistance, as their transaction workflows may expose information that could potentially be exploited under certain conditions.

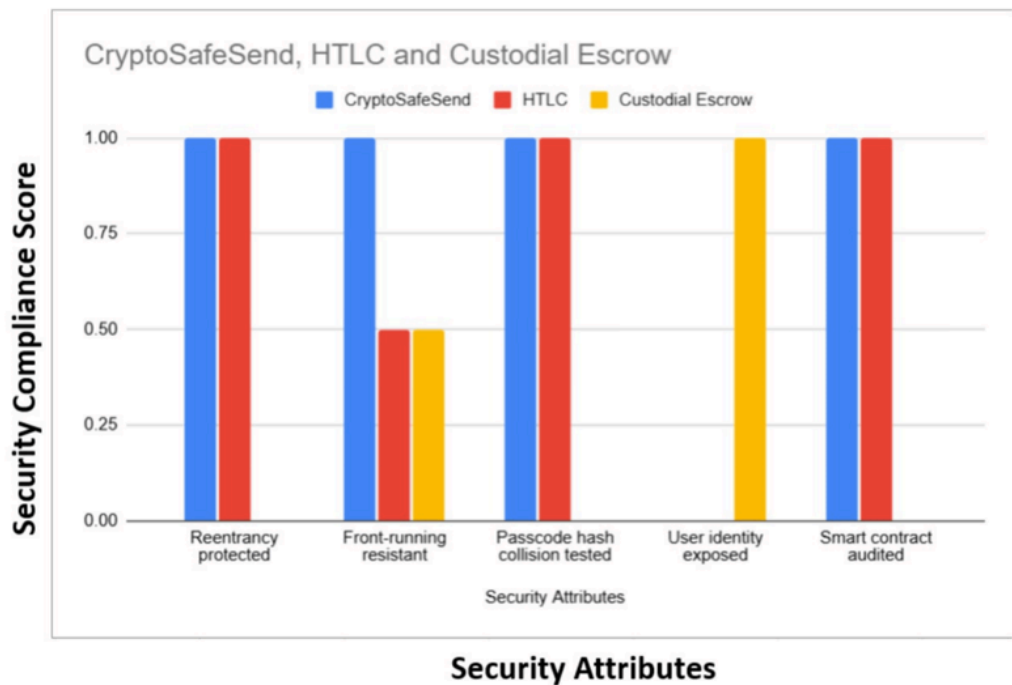
Passcode hash collision resistance ensures that cryptographic hashing mechanisms prevent two different inputs from producing the same hash output. CryptoSafeSend achieves full compliance by utilizing the Keccak-256 cryptographic hash function, which provides strong collision resistance and secure verification. HTLC also demonstrates full compliance through cryptographic hash-based locking mechanisms. Custodial escrow systems, however, do not explicitly implement or verify collision-resistant cryptographic validation.

User identity exposure is an important privacy consideration in decentralized systems. CryptoSafeSend and HTLC preserve user privacy by relying on blockchain address-based authentication without exposing personally identifiable information. In contrast, custodial escrow systems typically require identity verification and centralized account management, which may expose sensitive user information and reduce privacy protection.

Smart contract auditing is essential for ensuring the security and correctness of decentralized applications. CryptoSafeSend and HTLC utilize transparent, verifiable smart contract logic that can be independently reviewed and audited, thereby improving trust and reliability. Custodial escrow systems do not provide the same level of transparency, as their internal security mechanisms are typically proprietary and not publicly verifiable.

How to cite this article:

Overall, the comparative analysis demonstrates that CryptoSafeSend provides strong security guarantees across multiple dimensions, including cryptographic integrity, privacy protection, and resistance to common blockchain attack vectors. These characteristics establish CryptoSafeSend as a robust and reliable framework for secure, decentralized asset transfer and escrow management.



The x-axis represents the evaluated security attributes, while the y-axis represents the security compliance score (1 = fully supported, 0.5 = partially supported, 0 = unsupported).

FIGURE 22: Security Attribute Comparison of CryptoSafeSend, HTLC, and Custodial Escrow Platforms.

Source: Author’s own work.
HTLC, Hashed Time Lock Contract

Conclusions

CryptoSafeSend preserves the trustless nature of blockchain systems while offering a safe and useful solution to inadvertent cryptocurrency transfers through the use of a conditional transfer mechanism based on cryptographic passcodes. A 7-day locking period was successfully implemented in this work, which enables the sender to recover or revoke funds prior to the transaction being finalized if the recipient fails to claim them using the correct passcode. The risks of malicious activity, incorrect addresses, and human error are greatly decreased by this time-bound safeguard. Minimal computational and gas overhead, avoidance of duplicate claims, and practical deployability within current smart contract frameworks are demonstrated by experimental validation in an emulated Ethereum environment. The findings demonstrate that CryptoSafeSend improves transaction security without sacrificing efficiency or decentralization.

To further increase scalability and cost-effectiveness, future extensions might support multi-token standards like ERC-20 and ERC-721, deepen integration with popular DeFi wallets, implement advanced time-locked recovery configurations, multi-factor recipient authentication, on-chain dispute resolution mechanisms, and deploy on Layer-2 or other EVM-compatible networks.

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

Appendices

```
function createTransaction(address _recipient,
bytes32 _codeHash) public payable {
    require(msg.value > 0, "Send some ETH");
    transactions[txCounter] = Transaction(
        msg.sender,
        _recipient,
        msg.value,
        _codeHash,
        false,
        false
    );
}
```

FIGURE 23: Function to Create a Transaction.

```
function claimFunds(uint _txId, string memory _code)
public {
    Transaction storage txn = transactions[_txId];
    require(!txn.claimed, "Already claimed");
    require(!txn.cancelled, "Transaction
cancelled");
    require(msg.sender == txn.recipient, "Not the
intended recipient");
    require(txn.codeHash ==
keccak256(abi.encodePacked(_code)), "Invalid
code");
    txn.claimed = true;
    payable(txn.recipient).transfer(txn.amount);
    emit Claimed( txId, msg.sender);}
```

FIGURE 24: Function to Claim Funds With a Correct Passcode.

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

```
// ===== REFUND TRANSACTION (AFTER 7 DAYS) =====
function refundTransaction(uint256 _txId) external {
    Transaction storage txn = transactions[_txId];

    require(!txn.completed, "Transaction already
completed");
    require(block.timestamp > txn.expiryTime, "Refund not
available yet");

    txn.completed = true;
    payable(txn.sender).transfer(txn.amount);

    emit TransactionRefunded(_txId, txn.sender);
}
```

FIGURE 25: Function to Refund a Transaction If It Is Unclaimed

```
function getTransaction(uint _txId) public view
returns (
    address sender,
    address recipient,
    uint amount,
    bool claimed,
    bool cancelled
) {
    Transaction memory txn = transactions[_txId];
    return (txn.sender, txn.recipient, txn.amount,
txn.claimed, txn.cancelled);
}
```

FIGURE 26: View the Transaction Performed by Transaction ID.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Kamalika Bhowal, Srijit Mondal, Kousik Dasgupta, Abir Chattopadhyay

Acquisition, analysis, or interpretation of data: Kamalika Bhowal, Srijit Mondal, Kousik Dasgupta, Abir Chattopadhyay

Drafting of the manuscript: Kamalika Bhowal, Kousik Dasgupta

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

Critical review of the manuscript for important intellectual content: Kamalika Bhowal, Srijit Mondal, Kousik Dasgupta, Abir Chattopadhyay

Supervision: Kousik Dasgupta, Abir Chattopadhyay

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue. **Animal subjects:** All authors have confirmed that this study did not involve animal subjects or tissue. **Conflicts of interest:** In compliance with the ICMJE uniform disclosure form, all authors declare the following: **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

Data Availability Statements

The program code and supporting implementation files for the CryptoSafeSend framework are openly available in GitHub at <https://github.com/srijitmondal/CryptoSafeSend>, reference CryptoSafeSend repository.

References

1. Swan M: [Blockchain: Blueprint for a New Economy](#). O'Reilly Media, Sebastopol, CA; 2015.
2. Janpitak N, Sathitwiriya Wong C: [Prevention of cryptocurrency loss from wrong receiver address](#). Journal of Information Science and Technology. 2021, 11:1-9. [10.14456/jist.2021.11](#)
3. Fröhlich M, Gutjahr F, Alt F: [Don't lose your coin! Investigating security practices of cryptocurrency users](#). DIS '20: Proceedings of the 2020 ACM Designing Interactive Systems Conference. 2020, 1751-1763. [10.1145/3357236.3395535](#)
4. Fröhlich M, Wagenhaus MR, Schmidt A: [Don't stop me now! Exploring challenges of first-time cryptocurrency users](#). DIS '21: Proceedings of the 2021 ACM Designing Interactive Systems Conference. 2021, 138-148. [10.1145/3461778.3462071](#)
5. Yu Y, Sharma T, Das S, Wang Y: [Don't put all your eggs in one basket: How cryptocurrency users choose and secure their wallets](#). CHI '24: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems. 2024, 1-17. [10.1145/3613904.3642534](#)
6. Hayama C, Kamidoi Y, Wakabayashi S: [Introduction of a new method for preventing recipient unapproved transactions to Bitcoin wallet](#). 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), Los Alamitos, CA, USA. 2022, 767-772. [10.1109/COMPSAC54236.2022.00126](#)
7. Kamidoi Y, Yamauchi R, Wakabayashi SI: [A protocol for preventing transaction commitment without recipient's authorization on blockchain and its implementation](#). IEEE Access. 2021, 9:24390-24405. [10.1109/access.2021.3056623](#)
8. Di Angelo M, Salzer G: [Identification of token contracts on Ethereum: standard compliance and beyond](#). International Journal of Data Science and Analytics. 2023, 16:333-352. [10.1007/s41060-021-00281-1](#)
9. Taherdoost H: [Smart contracts in blockchain technology: a critical review](#). Information. 2023, 14:117. [10.3390/info14020117](#)
10. Tsankov P, Dan A, Drachsler-Cohen D, Gervais A, Bünzli F, Vechev M: [Securify: practical security analysis of smart contracts](#). CCS '18: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018, 67-82. [10.1145/3243734.3243780](#)
11. Krishna Sree SB, Jeyashruthi M, Pavithra J, Uma S: [Blockchain with client assistance with the transaction reason](#). 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India. 2023, 1433-1438. [10.1109/ICPCSN58827.2023.00229](#)
12. Abramova S, Voskobojnikov A, Beznosov K, Böhme R: [Bits under the mattress: Understanding risk perceptions and security behaviors of crypto-asset users](#). CHI '21: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. 2021, 1-19. [10.1145/3411764.3445679](#)

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>

13. Mai A, Pfeffer K, Gusenbauer M, Weippl E, Krombholz K: [User mental models of cryptocurrency systems - a grounded theory approach](#). SOUPS'20: Proceedings of the Sixteenth USENIX Conference on Usable Privacy and Security. 2020, 341-358.
14. Guan MY, Yu Y, Sharma T, Huang MZ, Qin K, Wang Y, Wang KY: [Security perceptions of users in stablecoins: Advantages and risks within the cryptocurrency ecosystem](#). 2025 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA. 2025, 2753-2771. [10.1109/SP61157.2025.00042](#)
15. Islam MN, Hossen MGS, Baidya SP, Emon MAU, Hossain MS: [A framework for tracing the real identity of a Bitcoin scammer](#). 2021 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2), Rajshahi, Bangladesh. 2021, 1-4. [10.1109/IC4ME253898.2021.9768636](#)
16. Soni DK, Sharma H, Bhushan B, Sharma N, Kaushik I: [Security issues and seclusion in Bitcoin system](#). 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT), Gwalior, India. 2020, 223-229. [10.1109/CSNT48778.2020.9115744](#)
17. Zamani E, He Y, Phillips M: [On the security risks of the blockchain](#). Journal of Computer Information Systems. 2020, 60:495-506. [10.1080/08874417.2018.1538709](#)
18. Yousaf H, Kappos G, Meiklejohn S: [Tracing transactions across cryptocurrency ledgers](#). SEC'19: Proceedings of the 28th USENIX Conference on Security Symposium. 2019, 837-850.
19. Chan WK, Chin JJ, Goh VT: [Evolution of Bitcoin addresses from security perspectives](#). 2020 15th International Conference for Internet Technology and Secured Transactions (ICITST), London, United Kingdom. 2020, 1-6. [10.23919/ICITST51030.2020.9351346](#)
20. Omote K, Suzuki A, Sato T: [A new method of assigning trust to user addresses in Bitcoin](#). 2020 Second International Conference on Blockchain Computing and Applications (BCCA), Antalya, Turkey. 2020, 38-43. [10.1109/BCCA50787.2020.9274463](#)
21. Froehlich M, Waltenberger F, Trotter L, Alt F, Schmidt A: [Blockchain and cryptocurrency in human-computer interaction: A systematic literature review and research agenda](#). DIS '22: Proceedings of the 2022 ACM Designing Interactive Systems Conference. 2022, 155-177. [10.1145/3532106.3533478](#)
22. Khan AG, Zahid AH, Hussain M: [Security of cryptocurrency using hardware wallet and QR code](#). 2019 International Conference on Innovative Computing (ICIC), Lahore, Pakistan. 2019, 1-5. [10.1109/ICIC48496.2019.8966739](#)
23. Rakdej P, Janpitak N, Warasart M, Lilakiatsakun W: [Coin recovery from inaccessible cryptocurrency wallet using unspent transaction output](#). 2019 4th International Conference on Information Technology (InCIT), Bangkok, Thailand. 2019, 99-103. [10.1109/INCIT.2019.8911915](#)
24. Christidis K, Devetsikiotis M: [Blockchains and smart contracts for the Internet of Things](#). IEEE Access. 2016, 4:2292-2303. [10.1109/access.2016.2566339](#)
25. Alruwaili A, Kruger D: [Intelligent transaction techniques for blockchain platforms](#). 2019 International Conference on Computing, Electronics & Communications Engineering (iCCECE), London, UK. 2019, 177-182. [10.1109/iCCECE46942.2019.8941837](#)
26. Bhowal K, Mondal S, Chattopadhyay A, Dasgupta K: [ElectraGuard: a blockchain-based opinion polling system for tamper-proof surveys](#). 2025 International Conference on Artificial Intelligence for Computing, Astronomy and Renewable Energy (AICARE), Kolkata, India. 2025, 1-6. [10.1109/AICARE66005.2025.11402767](#)
27. Froehlich M, Hulm P, Alt F: [Under pressure: A user-centered threat model for cryptocurrency owners](#). ICBT '21: Proceedings of the 2021 4th International Conference on Blockchain Technology and Applications. 2021, 39-50. [10.1145/3510487.3510494](#)

How to cite this article:

Bhowal K, Mondal S, Dasgupta K, et al. (April 09, 2026) CryptoSafeSend: A Blockchain-Based Secure Transaction Mediator for Decentralized Finance. Cureus J Comput Sci 3 : es44389-026-00054-6. DOI <https://doi.org/10.7759/s44389-026-00054-6>