Framework for privacy preserving credential issuance and verification system using soulbound token

Siddhant Reddy[1*], and *Dharmender* Singh Kushwaha[1†]

[1]Motilal Nehru National Institute of Technology, Department of CSE, Allahabad, India

Abstract. This paper proposes a framework for privacy-preserving credential issuance and verification over the public blockchain. The credential used in this framework is a soulbound token (SBT), a non-transferrable non-fungible token (NFT) verifiable on the blockchain. Once the issuing organization issues the credential, this framework gives the holder complete control of the credential. This privacy-preserving property allows the holder to selectively disclose the credential attributes in the verification process. The framework proposed suggests a decentralized credential recovery mechanism if the credential holder loses their private key. This paper compares this framework's efficiency with different schemes based on privacy-preserving, selective disclosure, and decentralized credential recovery. This paper also compares the overhead for credential issuance and verification with Merkle trees. This paper also discusses the real-world use cases where this framework can be applied.
Keywords: Public Blockchain, Soulbound Token, Non-Fungible Token, Decentralize Recovery, Privacy-Preserving Credential.

1 Introduction

Blockchain technology is a decentralized, distributed database that records a continuously growing list of transactions, called blocks, across a network of computers. Every blockchain block has a timestamp and a link to the previous block. The blockchain cannot be altered retroactively without altering all subsequent blocks and the network's consensus. It was initially developed to enable secure, transparent, and tamper-proof record-keeping for the digital currency Bitcoin [1]. Blockchains are often used to create transparent and secure systems for recording and verifying transactions, such as financial transactions or the transfer of assets. Blockchain has improved traceability and visibility in the supply chain management [2] sector. In the public sector [3], blockchain technology considerably improves the transparency and efficiency of government operations. In addition, blockchain technology has the potential to revolutionize several industries, including finance [4] and even voting systems [5], by enabling secure and transparent record-keeping and reducing the need for intermediaries.

* Siddhant Reddy: siddhant.2021is16@mnnit.ac.in
† Dharmender Singh Kushwaha: dsk@mnnit.ac.in

This distributed nature makes it more resistant to fraud and tampering, as it would require a large number of nodes to conspire to alter the record of transactions. One important characteristic of blockchain technology is decentralization. It means that any single entity or organization does not control it. It instead uses a network of computers called nodes to verify and log transactions. This characteristic of blockchain increases security, as there is no single point of vulnerability.

Further technical advancement in blockchain technology leads to Ethereum [6] and Smart contracts [7]. Ethereum was developed in 2014 by Vitalik Buterin, a decentralized, open-source blockchain platform that enables smart contracts, which are computer programs that run exactly as intended without the risk of fraud, censorship, interruption, or outside interference. These contracts are stored on the Ethereum blockchain and executed by the Ethereum Virtual Machine (EVM), a decentralized, Turing-complete virtual machine that can execute arbitrary code. Ethereum has become one of the leading blockchain platforms for developing decentralized applications (Dapps).

A smart contract is a self-executing term with an agreement between multiple users inserted straight into lines of code. The agreements in the code are executed by the blockchain network and are transparent, traceable, and irreversible. Smart contracts make it possible to automate the execution and enforcement of contracts, saving time and reducing the potential for errors and disputes. Smart contracts on the Ethereum platform are written in Solidity, a programming language specifically designed to develop smart contracts.

NFT [8] stands for non-fungible token. It is a digital asset representing ownership of a unique digital item, as opposed to cryptocurrencies like Bitcoin [1], which are fungible tokens that can be exchanged for other tokens of the same type. NFTs are typically built on blockchain technology, which allows for the creation of unique, verifiable digital assets, and their ownership is recorded on the blockchain, making it easy to track and verify ownership. NFTs can be a piece of art, a collectible, a credential, or even a tweet.

NFT metadata is information about NFT that is used to identify and verify the authenticity of the NFT. This metadata may include information about the NFT's creator, the date it was created, its title or name, and other relevant details about the NFT. NFT metadata ensures that the NFT is unique and cannot be replicated or counterfeited, which is crucial for building trust in NFT. Apart from this, NFT metadata can also include details about the item or asset the NFT represents, such as an image of digital art, video, audio clip, and other relevant information about the NFT. NFT metadata can be stored in two ways, on-chain and off-chain. On-chain metadata is stored directly on the blockchain. Because it is stored on the blockchain, on-chain metadata is secure and cannot be altered or deleted without a trace. On the other hand, off-chain metadata is stored outside of the blockchain, typically on a centralized server or database. While this allows for greater flexibility and easier access to the metadata, it also means that the information is not as secure and may be more vulnerable to tampering or corruption. In decentralized storage or file-sharing systems like Interplanetary File System (IPFS) [9] to store this metadata, then this makes it tamper-resistant. In the context of NFTs, IPFS is often combined with on-chain metadata to create a complete and verifiable record of an NFT.

Using NFT as a credential issuance and verification system will have a significant drawback due to its transferability. To overcome this drawback Vitalik Buterin and co-authors propose a solution that will make the NFT non-transferable by design. They introduced a concept of Soulbound token (SBT) [10]. Soulbound token (SBT) is Non-transferrable NFT (probably revocable in certain conditions) that shows commitments, credentials, and affiliations. SBT is publicly visible and verifiable on the blockchain. The wallet that issues soulbound tokens is called Souls. Smart contract can be use to program recovery processes for soulbound tokens to reissue the SBT when the private key is lost and compromised. SBT can solve the problem of credential issuance on the blockchain.

Despite having this advanced technology, we still use paper-based credentials in some aspects of our life, like passports, driving licenses, etc. These paper-based credentials have many disadvantages, which include inefficiency, lack of portability, limited accessibility, impact, limited security, etc. The paper-based system is also prone to credential fraud, which is using fake or stolen credentials to gain access to sensitive information or services protected by a certificate-based authentication system.

Nowadays, some organization has shifted to digital credentials. This digital credential does solve some of the problems that paper-based credentials possess, but it still needs to solve all the problems. The digital credential system requires a central party to manage them, significantly affecting users' privacy. This needs to give complete data control to the holder of the credential. Due to centralization, user information can be compromised if any unauthorized entity or malicious group breaks the security.

A blockchain-based solution for credential sharing can bring many benefits: the credentials or their fingerprints stored and shared on the blockchain are tamper-proof; a malicious user cannot alter any documents stored on-chain. The decentralized and immutable nature of blockchain provides participants with a trusted, neutral credential-sharing platform and undeniable evidence of all recorded transactions.

2 Related work

The concept of soulbound token was introduced by Vitalik Buterin, E. Glen Weyl, and Puja Ohlhaver in the paper called decentralized society [10]. They define Soulbound tokens (SBTs) as non-transferable tokens representing commitments, credentials, and affiliations. The accounts or wallets that hold SBT are called Souls. The non-transferability of SBTs shows a relationship of trust rather than transferable assets and commitments in the current web 3.0 space. The souls can issue SBTs to other souls, showing the credential's commitment and authenticity. The non-transferability of the SBT will create a problem for losing SBT in case of private key loss. This problem can be solved by using a recovery paradigm. The original paper suggests using social recovery that relies on the trusted relationship of the SBT owner. The more advanced recovery process is community recovery which relies on the holder's social network, such as communities. Examples of these communities can be employers, universities, events, etc. The recovery process requires the consent of community members to recover the private key or reissue SBT. The authors further talk about private souls because on-chain SBTs will make the information public for applications. This issue can be solved by storing the data off-chain. Storing off-chain data can be any secure cloud service or decentralized data storage system like Interplanetary File System (IPFS) or Filecoin [11]. The off-chain data storage gives complete control of data to the SBT owner. The owner can decide who and when to show the SBT to other souls. The off-chain storing of SBT data can allow the owner's soul to reveal data selectively to other souls using cryptography. This can extend to zero-knowledge-proof [12] to provide more privacy. The SBTs can be used to claim false identities or commitments, which can be easily identified by checking the issuer's wallet address. SBTs share similarities with verifiable credentials (VCs) [13], a credential standard developed by W3C. VCs provide more privacy-preserving credential systems than SBTs because of the public availability of credential data. However, VCs have needed help with the recovery process, like social and community recovery, which is possible for SBT. Another term discussed in the paper is revocability. The issuer can burn A revocable SBT or reissue it to a new soul. The revocable SBT makes sense when the wallet keys are lost or compromised. This characteristic can issue timebound SBT to souls representing subscription or affiliation for a particular time period. The revocability and recovery process works simultaneously.

Significant research has yet to be on soulbound tokens (SBTs) [10] as it is very recently

introduced. In the whitepaper for soulbound tokens, Vitalik talks about how we can leverage these token characteristics in the current system. This paper [14] discusses the future of soulbound tokens in web 3.0 and how they differ from the verifiable credentials [13] in the self-sovereign identity field. It shows the challenges we can face due to the non-transferability of soulbound tokens. This thesis [15] imagines a world with Self-Sovereign Identity (SSI), which focuses on bringing the ownership of credentials to the holder. They suggest merging SSI with off-chain credentials and on-chain smart contracts. This KYC-compliant identity scheme [16] talks about doing KYC in exchanges and authenticating the user with the soulbound token. This soulbound token is linked with an identity with the authorized organization. This paper [17] talks about how to use blockchain technology that enhances the digital economy of the system. This NFT paper [18] shows how to use emerging NFT technology in business and organization processes.

In the original paper [10], Vitalik and co-authors suggested that multi-sig recovery can have difficulties with security and transaction. The other solution is social recovery, where a user (holder of SBT) can allow other wallets to be the guardian wallets. These guardian wallets then initiate the recovery process on the user's behalf, as the user knows the identity of the wallets in real life. This recovery process security depends mainly on the number of guardians the algorithm allows the smart contract to recover SBT. Too many or very few guardians can lead to a bad recovery algorithm. Vitalik also suggests a broader and similar process called community recovery, where the holders of SBT in the social network or community will be responsible for the recovery of the SBT. The Cerberus [19] uses multiple addresses to form a revoking process for the credential. These addresses are associated with the credential owner at the time of the credential issuance. This revoking process can be seen as halfway through the recovery process for soulbound tokens. This paper [14] shows the challenges one might face while implementing the recovery process for soulbound tokens. The recovery process should be carefully implemented to reach the full potential of the token.

Recently, various blockchain systems have provided a document and credential verification solution. In the paper [20], the author suggests a Go chain (government blockchain) to authenticate documents that minimize document fraud. This system will make the current digital document system more efficient and secure. The document verification system proposed in this paper [21] is tamper-proof and ensures trust. This system is a more efficient, secure, and cost-effective mode of document verification for employers. Cerberus [19] is implemented to fight current real-world fraud scenarios. This solution also suggests the credential revocation process and is also good at data privacy and verification to remove credential forgery. The CredChain [22] presents a self-sovereign identity [23] architecture for secure issuance, sharing, and verification of credentials. In another architecture [24] author ensures the credential verification system and suggests a privacy control mechanism. The CredenceLedger [25] uses permissioned blockchain and cryptography techniques to make an easily verifiable credential system for the education sector. Another system for an academic record is proposed in this paper [26] that reduces wait time for credential transfer. This uses Hyperledger [27], which is scalable and cost-effective. A novel protocol [28] is presented that is completely user-centric and privacy-preserving. This [29] paper suggests methods to overcome over-centralization. This gives complete responsibility for the integrity and stored information and verification to the user in a blockchain-based environment. A solution called Coconut [30] is a library for Chainspace and Ethereum, which allows a subset of attributes visibility to the verifier and ensures confidentiality and authenticity even when the issuing authorities are malicious. This paper [31] discusses the challenges of using blockchain in education.

This paper [30] divides the degree information into two sets and provides complete control to the user on which part to disclose to the verifier. This solution also does not disclose the credential against the user's will. The solution proposed in this [29] research paper is Merkle trees [32]. Another paper [24] uses Merkle trees to disclose information anonymously

while having complete control over the credential. This paper [22] uses GGM (Goldreich-Goldwasser-Micali) and Merkle hash tree to implement selective disclosure in the system. This paper [30] uses zksnarks for securing credential privacy and provides users to selectively disclose information on-chain using the smart contract. This paper [16] uses the BBS+ algorithm to prove an individual's identity to the supervisor's entity, which does not give more information than is needed. The BBS+ algorithm makes use of zero-knowledge proof [12] to prove something to the verifier without revealing the actual information.

None of the above mention systems use soulbound tokens to issue credentials on the chain. The suggested framework is doing privacy-preserving disclosure of credentials which is under complete control of the credential holder. Also, suggesting a recovery process that needs to be fully centralized and uses the current system architecture effectively.

3 Methodology

The solution proposes the credential issuance, verification, and recovery system using soulbound tokens on the blockchain. This solution provides users complete control over the credential information with selective disclosure. The credential issued are on blockchain in the form of NFT, and the data of the credential (i.e., NFT) is stored on IPFS (Interplanetary file system) in an encrypted format.

3.1 System architecture

The proposed solution implementation includes three entities; the user (holder), the issuer entity, and the verifier. The organization can be any trusted organization or government department that manages and issue the SBT to the user. The user is an ordinary individual who requests the organization for credential issuance. The verifier can be individuals or organizations that desire credential verification. The system architecture for the credential issuance and recovery process is shown in Fig. 1.

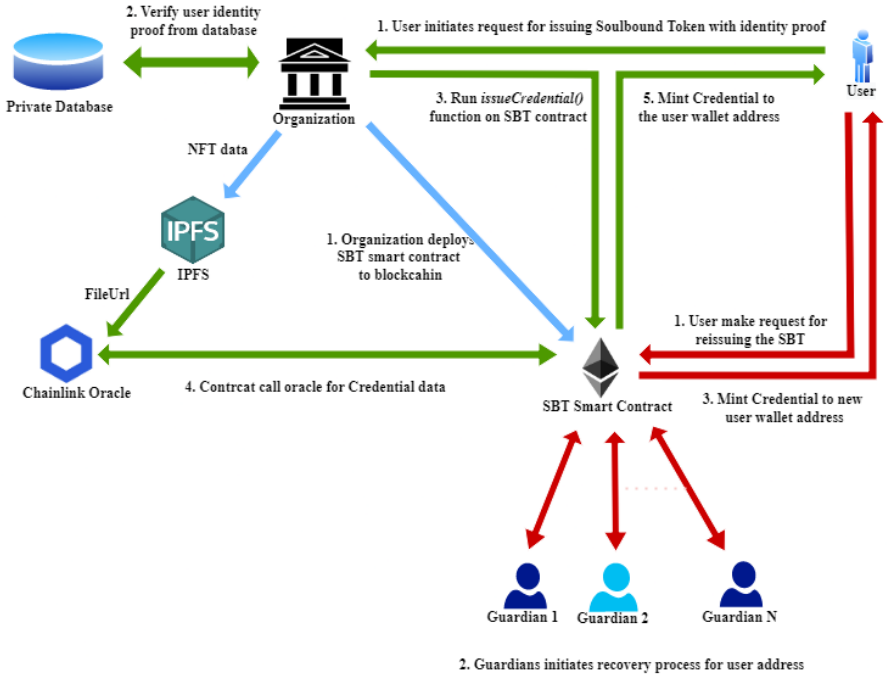

Fig. 1. System architecture for credential issuance and recovery process.

3.2 Program flow

The flow of the system is described in the following steps.

3.2.1 Deploying contract

The issuing organization deploys the SBT contract to the blockchain, which any entity can access.

3.2.2 Issuing of credential

The user initiates the request for issuing the credential. The user gives the user identity, wallet address, and address of guardians to the organization. The organization then verifies the user identity information and checks for valid guardian addresses, which must not be duplicated. After successfully verifying the data, the organization runs the *issueCredential()* function on the SBT contract. The *issueCredential()* function works in the following steps (as shown in Fig. 2).

The *issueCredential()* function will call chainlink oracle for off-chain data. The off-chain function generates credential attributes and encrypts them with a user public key. After generating the encrypted credential, the off-chain function uploads this credential on IPFS and gives the file URL to the oracle, which returns it to the SBT contract.

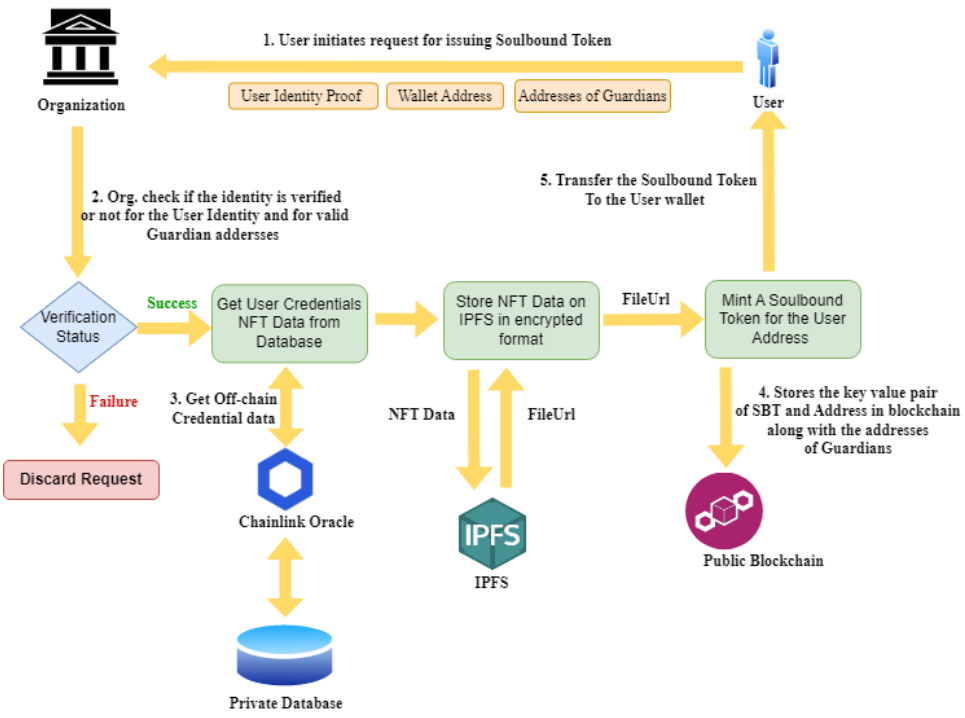

Fig. 2. Program flow for the organization and the user for issuing the soulbound token credential.

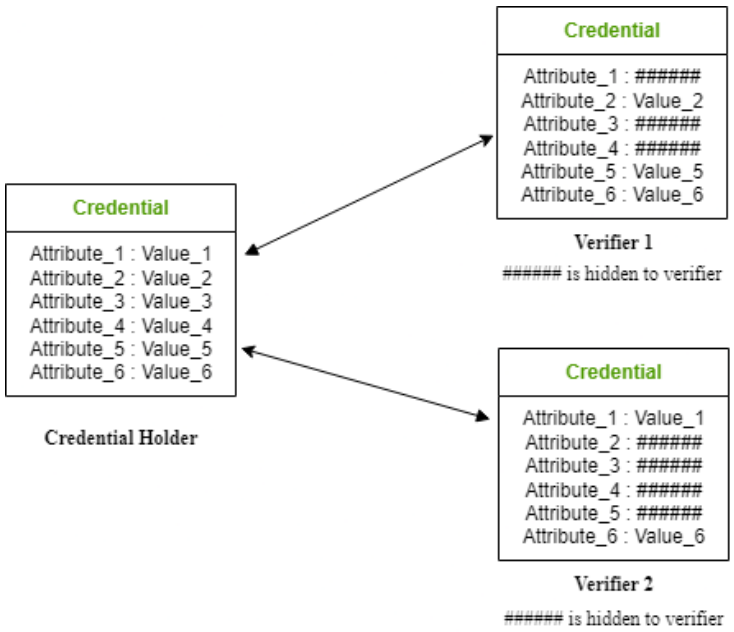

Fig. 3. Selective disclosure of credential for different verifiers.

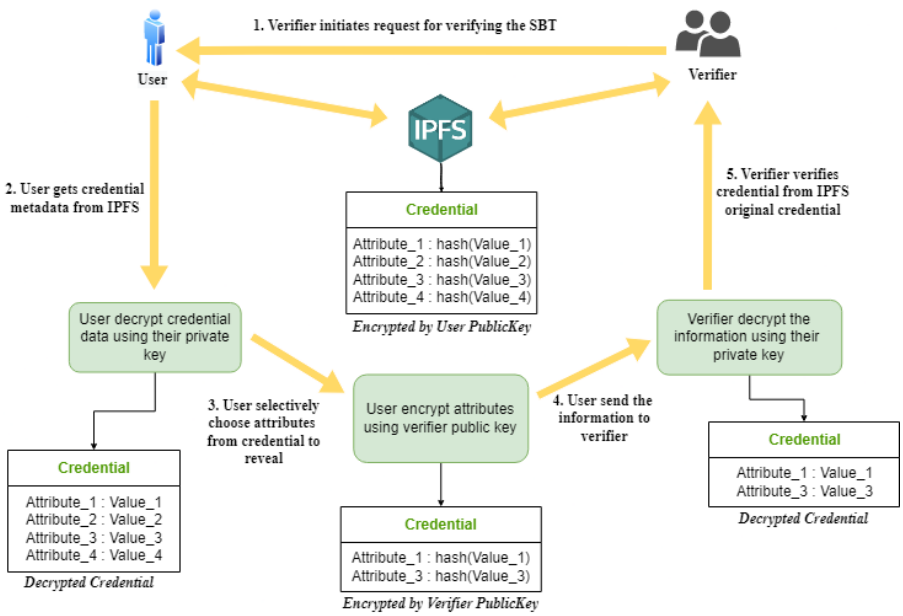

Fig. 4. Program flow for the User and Verifier for verifying the soulbound token credential.

3.2.3 Verification of credential

This scheme provides selective disclosure of credential to the verifier. The holder of credential has complete control over the information to show to the verifier. The selective

disclosure of credential to different verifier is show in the Fig. 3.

The verification process starts with the verifier requests the SBT credential from the holder. The credential metadata is stored on IPFS encrypted by the holder public key so this can only be seen by holder after decrypting it with the private key of the holder only. The holder can selectively choose to disclose the attributes to the verifier. The holder first decrypts the attributes of credential with the private key. After that holder encrypt it with the verifier public key and send it to the verifier. The verifier decrypts the information send by the holder. To checks the authenticity of the credential information, send by the holder verifier checks the hash of the decrypted information with the holder public key with the original stored credential on the IPFS. Hence verifier successfully authenticates the credential. The different state of credential in verification process is shown in Fig. 4.

3.2.4 Recovery Process for SBT

As shown in Fig. 5, The user initiates the *reissueCredential()* function with the new wallet address. The credential recovery requires guardians to call the *initiateRecovery()* function on SBT contract. If the function is called more the minimum threshold times, then the *executeRecovery()* function will get executed, which will burn the previous SBT tied to the old address and issue the new SBT to the new user wallet address.

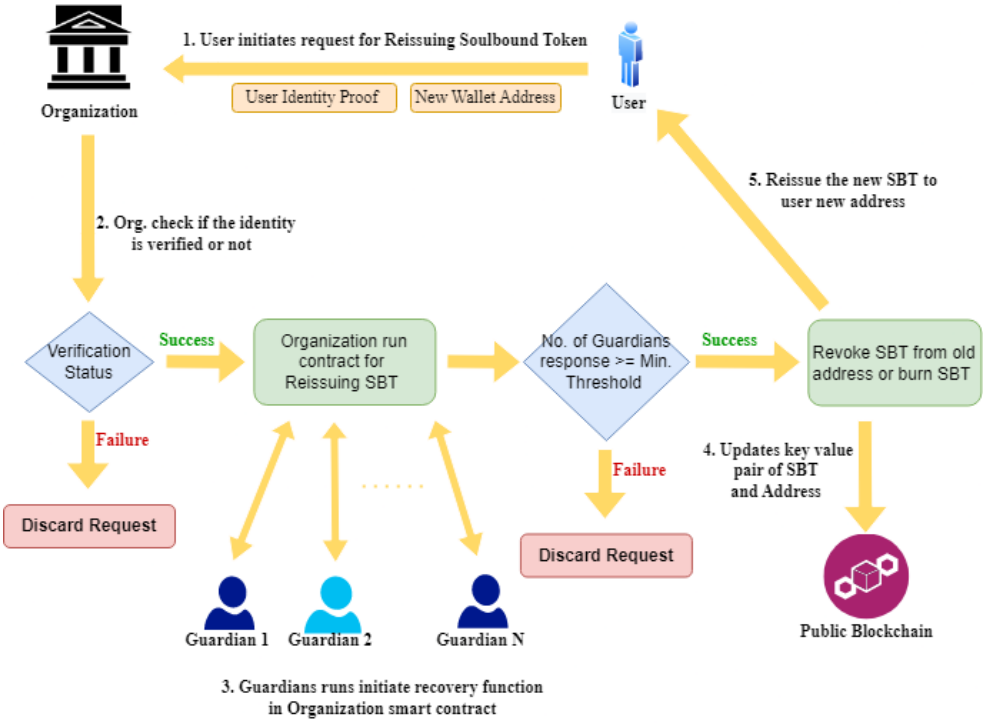

Fig. 5. Program flow between the organization and the user for reissuing the soulbound token credential

3.3 Smart contract

This scheme uses one smart contract credentialSBT. This smart contract is written on solidity. The structure of credentialSBT is shown in Fig. 6.

The function *initiateCredentialRequest()* runs after the organization validates the user

identity off-chain. This takes the user address and an array of guardian addresses as input. The guardian addresses are the addresses who want to be the guardian for the user address. This function checks for the validity of guardian addresses and then executes the *issueCredential()* function.

This *addGuardian()* function takes the user address and their respective guardian addresses array as input and adds the guardian addresses hash to the respective mapping with the user address. Before adding to the mapping, it checks for duplication in guardian addresses.

The *issueCredential()* function took the user address as input and used it to issue the SBT credential to the user address. This function works with several function calls. First it calls a *getCredentialUrl()* function, which reaches an off-chain API using a chainlink oracle that searches the credential in the organization private database. This API encrypts the credential attributes using the user public key and uploads it to the IPFS. The generated file URL is returned to the *issueCredential()* function. After getting the credential file URL from the function, it calls the *safeMintCredential()* function to mint the soulbound token for the token URL and assign it to the use address. At last, it store the tokenId to the respective user address in the addressToSBT mapping.

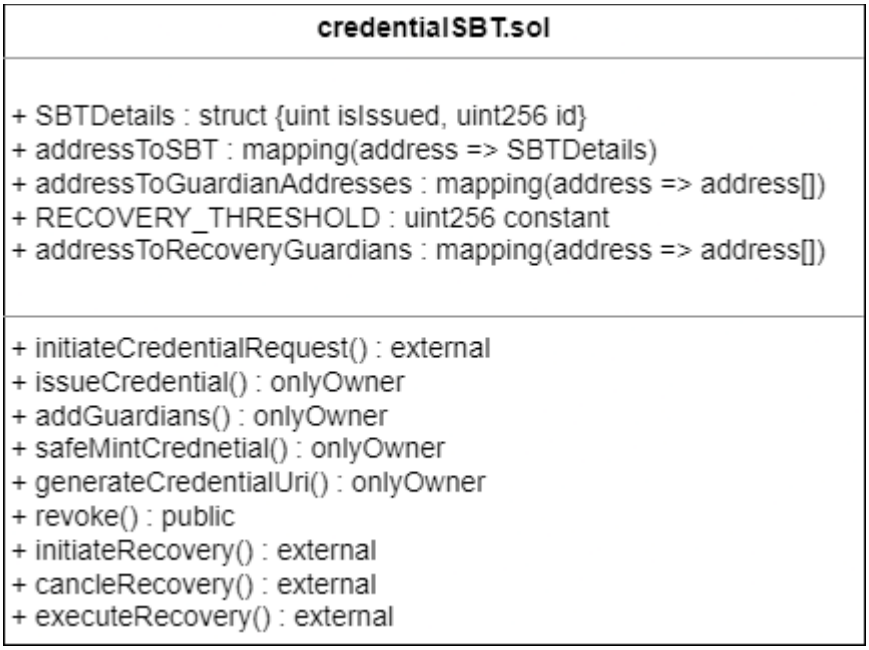

Fig. 6. CredentialSBT.sol smart contract structure for the scheme

The *revoke()* function burns the tokenId for the user address. This function runs when someone call the recovery process for reissuing the SBT.

The *initiateRecovery()* function allows a guardian to initiate the recovery process for the user. It first checks the validity of the guardian address concerning the user address and whether the guardian address has already started the recovery process. At last, it add the guardian address to the addressToRecoveryGuardians mapping for the user address.

The *cancleRecovery()* function removes the guardian from the recovery process for the user. It first checks the validity of the guardian address for the user address and whether the guardian address has already initiated the recovery process. Then for valid requests, it removes the guardian address to the addressToRecoveryGuardians mapping for the user

address.

The *executeRecovery()* function takes the user address (i.e., old user address) and the new user address where the user wants to recover their SBT credential. This function first checks that the number of guardians must be more than the RECOVERY THRESHOLD for reissuing SBT. If this is true, it revokes the old SBT from the user address, mints the new SBT to the new user address, and reinitializes the addressToRecoveryGuardians array as the recovery process has finished.

4 Comparison and result

This section compares the different schemes (as shown in Table I) present based on three properties privacy-preservation, selective disclosure, and decentralized recovery process.

Table 1. Comparison of different schemes

Name of the Scheme	Privacy Preserving	Selective Disclosure	Decentralized Recovery Process
A Blockchain-based framework for secure Educational Credentials [33]	No	No	No
Blockchain-based Verifiable Credential Sharing with Selective Disclosure [22]	Yes	Yes	No
Cerberus: A Blockchain-Based Accreditation and Degree Verification System [19]	Yes	Yes	No
CredenceLedger: A Permissioned Blockchain for Verifiable Academic Credentials [25]	Yes	No	No
Coconut: Threshold Issuance Selective Disclosure Credentials with Applications to Distributed Ledgers [30]	Yes	Yes	No
A novel credential protocol for protecting personal attributes in blockchain [28]	Yes	Yes	No
Blockchain-based Learning Credential Verification System with Recipient Privacy Control [24]	Yes	No	No
A Permissioned Blockchain-Based System for Verification of Academic Records [26]	No	No	No
Our scheme using Soulbound token	**Yes**	**Yes**	**Yes**

The Chart (shown in Fig. 7) represents the number of hashes calculation require for generating the credential for the selective disclosure process. This paper compare our scheme that uses simple public cryptography with the merkle tree algorithm for selective disclosure.

The chart (shown in Fig. 8) shows the number of hashes calculation require for verifying the credential for the selective disclosure process. Our scheme takes constant time while the merkle tree algorithm has time complexity of *Log(N)*.

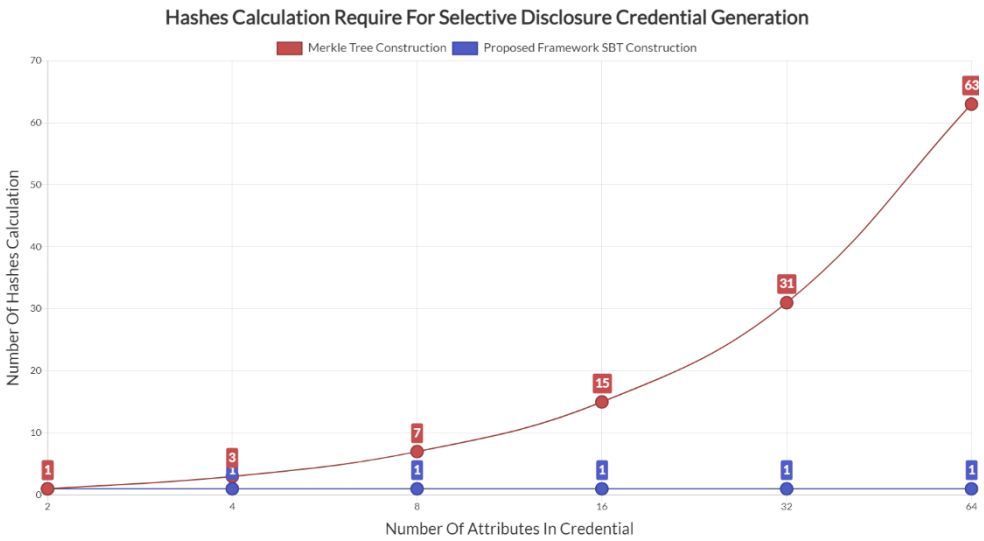

Fig. 7. Comparison between merkle tree and selective disclosure credential generation

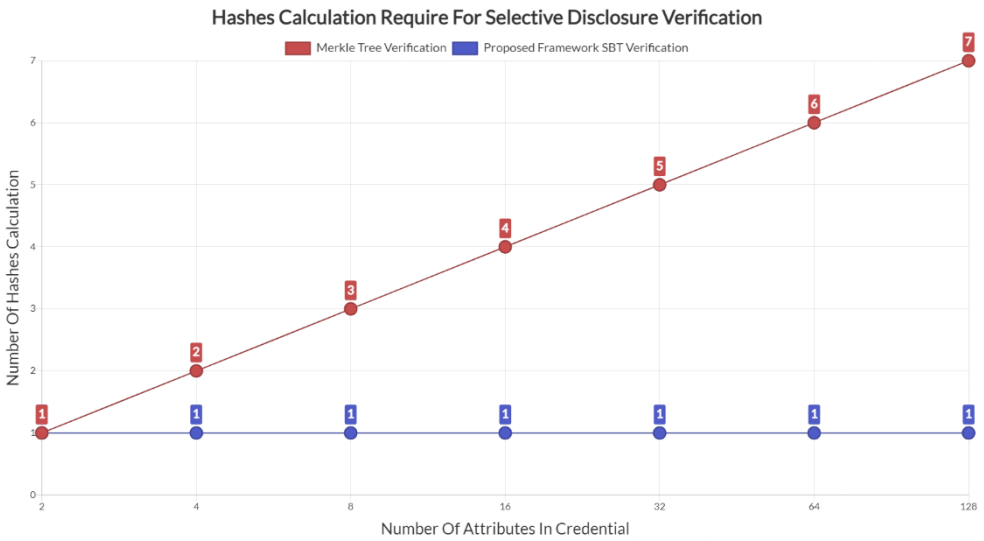

Fig. 8. Comparison between merkle tree and selective disclosure credential verification.

5 Privacy and security analysis

5.1 Privacy analysis

One of the main characteristics of blockchain is transparency. Transparency makes the blockchain decentralize as the ledges are publicly visible. Transparency can be a drawback in the credential system as the credential contains the user's private information. No one wants to reveal their personal information to everyone. For this reason, this framework choose to store our data on other decentralized file storage (i.e., IPFS). The credential data (NFT metadata) is encrypted using the holder public key so no one can see it. Apart from encrypting data, it also implement selective disclosure in credential sharing and verification,

which gives the holder complete ownership of the credential. The credential holder has full control over whom they share information and what they share. This data protection will preserve user privacy more than the current systemsSecurity analysis.

5.2 Security analysis

There are two biggest security problems in the blockchain wallet system, they are private key infringement and private key loss. If there is a private key infringement, the user can request his guardians to initiate the recovery process, and they can retrieve their credentials to the new wallet address. This will stop any malicious user from using the credential in the wrong way. If the user loses their private key, in that case, the user initiates the recovery process with the help of guardians and reissues that credential to the new wallet address.

The other threat to the user is that guardians can team up to reissue SBT to other wallets for their benefit. This requires enough guardians to initiate the recovery process. The coordination between the guardians can only be possible if they know other guardians' identities in the real world. The identities can be derived from the guardian's addresses stored on the chain. This threat can be solved by storing the hashes of the guardian's addresses on the blockchain. This will hide the actual addresses of the guardians. Hence it will be impossible for guardians to know each other in the real world and use the SBT for selfish purposes.

6 Conclusion and future work

Our proposed framework for a credential verification system on the blockchain can provide a secure, reliable way to verify the authenticity of credentials and prevent fraud. A credential verification system can create a decentralized and tamper-proof database of credentials, which can be accessed and verified by authorized parties transparently and securely. In addition, a credential verification system on the blockchain can enable real-time verification of credentials, which can help organizations and individuals to make more informed decisions and avoid the risks associated with credential fraud. Overall, using a credential verification system on the blockchain can provide significant benefits in terms of security, reliability, and efficiency. There are many use cases where this framework can be use to solve real-world problems.

This framework can solve various organizations' tedious repeating KYC (know your customer) verification problems. The problem is that one needs to do KYC for all organizations in the initial registration stage. One possible solution is to issue a soulbound token to the individual after completing the KYC. Organizations can then use this SBT of completeness to verify the person's identity on the blockchain. One possible implementation can be seen in Binance (popular cryptocurrency exchange), which introduces BAB (Binance account bound token) to users who complete the platform's KYC process.

The other use case is in storing personal credentials or documents (i.e., passport, driving license) in the blockchain. Due to the data privacy property of the framework, this framework can make the documents invisible to the public. Blockchain technology will stop credential forgery and remove unauthenticated documents from the server.

One can leverage this framework in the hiring process for companies. Many job openings require the employee's previous work history to ensure they are capable of the position. The company can issue an SBT for every job or every achievement an employee does while working for them. These SBTs can be verified by anyone and from anywhere in the world. Blockchain technology will reduce fake experience claims, increase efficiency, and reduce friction in the current system.

References

1. S. Nakamoto. *Bitcoin whitepaper*. URL: https://bitcoin.org/bitcoin. (2008).
2. M. M Queiroz, R. Telles, and S. H Bonilla. *Blockchain and supply chain management integration: a systematic review of the literature*. Supply Chain Management: An International Journal (2019).
3. M. H¨olbl, M. Kompara, A. Kamiˇsali´c, and L. Nemec Zlatolas. *A systematic review of the use of blockchain in healthcare*. Symmetry, 10(10):470 (2018).
4. M. Raikwar, S. Mazumdar, S. Ruj, S. Sen Gupta, A. Chattopadhyay, and K. Lam. *A blockchain framework for insurance processes*. In 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), pages 1–4. IEEE, (2018).
5. Kshetri and Voas, 2018 N. Kshetri, J. Voas. Blockchain-enabled e-voting. IEEE Software, **35** (4) (2018), pp. 95-99.
6. V. Buterin et al. A *next-generation smart contract and decentralized application platform*. white paper, 3(37):2–1 (2014).
7. Z. Zheng, S. Xie, H. Dai, W. Chen, X. Chen, J. Weng, and M. Imran. *An overview on smart contracts: Challenges, advances and platforms*. Future Generation Computer Systems, 105:475–491 (2020).
8. Q. Wang, R. Li, Q. Wang, and S. Chen. *Non-fungible token (nft): Overview, evaluation, opportunities and challenges*. arXiv preprint arXiv:2105.07447 (2021).
9. J. Benet, *IPFS-content addressed versioned P2P file system*, (2014), [online] Available: https://arxiv.org/abs/1407.3561.
10. E GlenWeyl, P. Ohlhaver, and V. Buterin. *Decentralized society: Finding web3's soul*. Available at SSRN 4105763 (2022).
11. J. Benet and N. Greco, *Filecoin: A decentralized storage network*, (2018), [online] Available: https://filecoin.io/filecoin.pdf.
12. U. Feige, A. Fiat, and A. Shamir. *Zero-knowledge proofs of identity*. Journal of cryptology, 1(2):77–94, (1988).
13. J. Sedlmeir, R. Smethurst, A. Rieger, and G. Fridgen. *Digital identities and verifiable credentials*. Business & Information Systems Engineering, 63(5):603–613, (2021).
14. Fe. Hildebrandt. *The future of soulbound tokens and their blockchain accounts*. In Konferenzband zum Scientific Track der Blockchain Autumn School (2022), number 2, pages 18–24. Hochschule Mittweida, (2022).
15. M. Casonato. *Owning your data through self-sovereign identity: agents implementation for verifiable credentials interaction* (2021).
16. N. Sun, Y. Zhang, and Y. Liu. *A privacy-preserving kyccompliant identity scheme for accounts on all public blockchains*. Sustainability, 14(21):14584, (2022).
17. C. Chen, L. Zhang, Y. Li, T. Liao, S. Zhao, Z. Zheng, H. Huang, and J. Wu. *When digital economy meets web 3.0: Applications and challenges*. IEEE Open Journal of the Computer Society, (2022).
18. S. Li and Y. Chen. *How non-fungible tokens empower business modelinnovation*. Business Horizons, (2022).
19. A. Tariq, H. Binte Haq, and S. Taha Ali. *Cerberus: A blockchain-based accreditation and degree verification system*. IEEE Transactions on Computational Social Systems, (2022).

20. I. Meirobie, A. Purna Irawan, H. Teja Sukmana, D. Putri Lazirkha, and N. Puji Lestari Santoso. *Framework authentication e-document using blockchain technology on the government system*. International Journal of Artificial Intelligence Research, 6(2), (2022).
21. K. Dhyani, J. Mishra, S. Paladhi, and I S. Thaseen. *A blockchain-based document verification system for employers*. In Proceedings of International Conference on Computational Intelligence and Data Engineering, pages 123–137. Springer, (2022).
22. R. Mukta, J. Martens, H. Paik, Q. Lu, and S. S Kanhere. *Blockchain-based verifiable credential sharing with selective disclosure*. In 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (Trust- Com), pages 959–966. IEEE, (2020).
23. A. M¨uhle, A. Gr¨uner, T. Gayvoronskaya, and C. Meinel. *A survey on essential components of a self-sovereign identity*. Computer Science Review, 30:80–86, (2018).
24. A. Mi San, N. Chotikakamthorn, and C. Sathitwiriyawong. *Blockchain-based learning credential verification system with recipient privacy control*. In 2019 IEEE International Conference on Engineering, Technology and Education (TALE), pages 1–5. IEEE, (2019).
25. R. Arenas and P. Fernandez. *Credenceledger: a permissioned blockchain for verifiable academic credentials*. In 2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), pages 1–6. IEEE, (2018).
26. A. Badr, L. Rafferty, Q. H Mahmoud, K. Elgazzar, and P. CK Hung. *A permissioned blockchain-based system for verification of academic records*. In 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), pages 1–5. IEEE, (2019).
27. C. Cachin et al. *Architecture of the hyperledger blockchain fabric*. In Workshop on distributed cryptocurrencies and consensus ledgers, volume 310, pages 1–4. Chicago, IL, (2016).
28. K. Singh, O. Dib, C. Huyart, and K. Toumi. *A novel credential protocol for protecting personal attributes in blockchain*. Computers & Electrical Engineering, 83:106586, (2020).
29. M. Ramachandran, N. Chowdhury, A. Third, J. Domingue, K. Quick, and M. Bachler. *Towards complete decentralised verification of data with confidentiality: different ways to connect solid pods and blockchain*. In Companion Proceedings of the Web Conference 2020, pages 645–649, (2020).
30. A. Sonnino, M. Al-Bassam, S. Bano, S. Meiklejohn, and G. Danezis. *Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers*. arXiv preprint arXiv:1802.07344, (2018).
31. A. Mohammad and S. Vargas. *Challenges of using blockchain in the education sector: A literature review*. Applied Sciences, 12(13):6380, (2022).
32. M. Szydlo. *Merkle tree traversal in log space and time*. In International Conference on the Theory and Applications of Cryptographic Techniques, pages 541–554. Springer, (2004).
33. S. Alam et al. *A blockchain-based framework for secure educational credentials*. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(10):5157–5167, (2021)