Privacy-Preserving Ownership Transfer: Challenges and An Outlined Solution Based on Zero-Knowledge Proofs

Mohammadtaghi Badakhshan
Electrical and Computer Engineering
University of Waterloo
Waterloo, Canada
mbadakhshan@uwaterloo.ca

Guang Gong
Electrical and Computer Engineering
University of Waterloo
Waterloo, Canada
ggong@uwaterloo.ca

2023 IEEE 9th World Forum on Internet of Things (WF-IoT) | 979-8-3503-1161-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/WF-IOT58464.2023.10539396

***Abstract*—Although employing blockchain in supply chain management (SCM) can provide benefits in numerous aspects such as traceability, transparency, and more, using public blockchain for SCM may compromise the privacy of the supply chain participants and their business secrets. In this paper, we review recent papers that integrate blockchain with SCM and the papers that propose privacy-preserving approaches for public blockchain. Then, we identify the problem in the existing solutions. Additionally, we present an outline of a framework that enables entities in a supply chain to upload their data records anonymously. This framework preserves unlinkability when transferring product ownership. The proposed scheme allows data auditors, who can be the end customers of a supply chain, to access a product's history and verify the authenticity of the data while preserving the privacy of the data uploader. We demonstrate that supply chain data records follow a directed acyclic graph (DAG), similar to the data structure that maintains data records in version control systems (VCS). Hence, this insight could make the framework applicable for anonymous version control systems based on blockchain.**

***Index Terms*—blockchain, ownership transfer, off-chain storage, directed acyclic graph, supply chain management, version control system, zero-knowledge proof, anonymity, unlinkability**

I. Introduction

A supply chain management (SCM) is a system used by a business and its suppliers to make and deliver certain products to the customers. A SCM consists of different components such as inventory management, transportation and logistics, and supply chain analytics. In SCM schemes, it is necessary to monitor the history of each product to gain visibility of the product's flow from producers to consumers. Data is either captured by the Internet of Things (IoT) sensors connected to products or entered manually. In the first case, the data is transmitted via different communication protocols, such as Radio-frequency identification (RFID) [1, 2], ZigBee sensors [3], or Bluetooth Low Energy (BLE) [4] to receivers. Then, the manually entered or sensor-captured data are aggregated for real-time analysis or future research. The collected data provides valuable information for business owners. Also, it eases the mind of the final customers about their product. For example, whether a wild-caught salmon is really caught from a lake, river, etc. or it is just a farm-raised salmon.

Applying blockchain technologies to secure SCM is a promising approach. Blockchain removes the need for trusting third parties; Moreover, the potential benefits that blockchain and IoT can give to SCM, such as traceability, transparency, less paper works and less code of conduct violation and fraud[5].

In supply chains, the ownership of products often changes as the products move through the chain. To accommodate this, an ownership model has been developed for RFID tags that enables the current owner to authenticate the tag and transfer its ownership. Numerous research works, such as [6, 7, 8], focus on transferring the ownership of RFID tags while maintaining privacy. However, the process of uploading measured data from the products still can potentially compromise the privacy of the product owners.

Supply chains' data records have a sequential format since they represent the conditions of a product over time. Furthermore, two distinct supply chains might merge due to various reasons, such as using the same warehouse, the same transportation, or assembly in the manufacturing stage. In contrast, a single supply chain can be divided into two or more sub supply chains for reasons like the distribution of a product to different locations. Consequently, Directed Acyclic Graphs (DAGs) are an optimal data structure for storing supply chain data records. For instance, a food supply chain can be seen as a DAG, where each node in the DAG signifies a data record captured by an entity responsible for moving, storing, or processing batches of food over a period of time.

DAGs are used in a range of applications where data flow needs to be depicted. The use of DAGs allows for mapping the relationship between different stages while considering the sequence and dependencies among these stages. Another example for an application of DAGs is in Version Control System (VCS) tools. Tools like Git [9] or Mercurial [10] rely on the DAG data structure to depict the evolution history of a project based on its commits.

Transferring applications such as SCM or VCS to permissionless (public) blockchains eliminates the need for trust in a manager and offers a range of benefits, which we will discuss later in this paper. However, by the nature of permissionless blockchain, miners, monitors, auditors and validators see the plain transactions on the blockchain. This could reveal the privacy of the data providers and actors in ownership transfer of the data from one to another. However, those will expose inventory information of different business sectors. So, for blockchain based SCM, the privacy of data uploader (anonymous problem), and ownership transfer of data (unlinkable problem) is a crucial factor to deploy this technology.

In this paper, our contributions include a comprehensive review of previous research on the use of blockchain technologies in SCM and privacy-preserving approaches in permissionless blockchains. We have identified issues prevalent in current schemes. Furthermore, we introduce an outline for a framework that provides a verifiable authentication mechanism. This is designed to maintain a shared DAG, ensuring the anonymity of data uploaders and unlinkability of data ownership transfers. The framework separates storage from authentication token management and uses these tokens to authorize data uploads to the storage. As a result, it achieves a higher level of anonymity and unlinkability. Additionally, it reduces the cost associated with data storage.

II. Problem Identification

A. Blockchain for Supply Chain Management

Centralized SCM softwares are controlled by one company to manage the products it owns. However, connecting those supply chains is necessary to keep the records of each product from production to consumption. It will be more vital when the products travel a long way to reach their destination or they are widely distributed [11]. since each business owns its own separated supply chain, collaborations between supply chains and also offering some excess resources in the supply chain of a business to another business is not straightforward [12]. Also, internationally collaborations is much harder due to different rules in each country.

Big companies, like Amazon, who owns their own supply chain let small retailers to use their platform to store, sell and ship their products. Small retailers cannot compete to have the same size SCM systems to track their products themselves as same as the big supply change management companies. Moreover, small retailers, logistics companies, etc. may not like to rely on existing SCM platforms, e.g., Amazon or IBM Supply Chain Intelligence Suite [13], since it might cost too much compared to their business size. Moreover, they might not like to share their business secrets with bigger companies. This violates their financial privacy.

SCM is one of the applications that are moving to blockchain-based platforms to decrease the need to trust the supply chain service providers and increase the fairness. Additionally, moving SCM to blockchain platforms can solve many of the abovementioned problems. According to the research by Saberi et al. [14], blockchain can eliminate the need for a trusted centralized organization to maintain five product dimensions: (1) the nature, (2) the quality, (3) the quantity, (4) the location, and (5) the ownership. Blockchain can provide a higher level of transparency and traceability to these dimensions, which helps verify authenticity, prevent fraud, and ensure ethical sourcing. These benefits lead to increased customer trust in products. Moreover, smart contract capability in some blockchains can automate various processes in blockchain as written rules which leads to reducing the paperworks and errors [15].

B. Blockchain for Version Control System

VCS is a software that tracks and manages changes made to files and code over time. It allows multiple developers to work collaboratively on a codebase without overwriting each other's changes. VCS also provides the ability to revert to previous versions of code, branch off to create separate versions, and merge changes back into the main branch.

Traditional VCS schemes are often centralized, with control held by a single entity. Github, a platform for software development and version control that uses Git, is an example of this; it is owned by Microsoft Corporation. Centralized VCSs have certain drawbacks, such as the potential for censorship, the risk of access being blocked, and the requirement for trust in the centralized manager to keep code repositories and files safe and unchanged. A relevant example of an incident involving a centralized VCS is the case of Tornado Cash, a decentralized privacy service that mixes cryptocurrencies [16]. The U.S. Treasury Department's Office of Foreign Assets Control (OFAC) blacklisted Tornado Cash [17], leading to GitHub removing it from their site. However, GitHub relist the code after several days. While Tornado Cash was primarily used for non-humanitarian purposes, this example highlights how a centralized manager for a VCS can block access to a public code repository.

Another important aspect of code development is the preservation of developers' anonymity. For instance, 'Satoshi Nakamoto' is the pseudonym used by the anonymous developer or developers of Bitcoin. Another example is Saeed Malekpour, a software developer who created a tool that simplified the uploading of photos to websites. Unbeknownst to him, this software was used by an adult content website. As a result, Malekpour was sentenced to death in Iran for allegedly designing and moderating adult content websites [18]. These two examples emphasize the importance of the privacy of software developers.

III. Related Works

A. Blockchain Based Supply Chain Management

Blockchain platforms can be divided into two categories: permissioned and permissionless blockchains. In permissioned blockchains, the number of participants is limited, all parties are known to each other, and there is no anonymity. IBM Food Trust [13] leverages Hyperledger Fabric [19] which is a permissioned blockchain and one of the Hyperledger projects hosted by The Linux Foundation. In Hyperledger Fabric, membership service provider (MSP) registers member who can publish and share information. Permissioned data access is an essential part of IBM Food Trust. Walmart collaborated with IBM in 2016 to build a blockchain-based system to trace the origin and transportation of food goods [20]. Their blockchain platform is also build on Hyperledger Fabric to trace over 25 products from 5 different suppliers. Blockchain technology has enabled Walmart to track a food item from the store back to its source within seconds. For instance, with the use of blockchain, the time required to trace the origin of mangoes in the U.S. has been reduced from 7 days to just 2.2 seconds [21].

On the other hand, permissionless blockchains are completely decentralized. This category of blockchains can provide a higher level of transparency and robustness. Also, they do not need a centralized party to grant or revoke permissions. Bitcoin and Ethereum are two well-known examples of permissionless blockchains. Permissionless blockchains can let any user participate as an entity in a supply chain. However, preserving the entities' anonymity while verifying the authenticity of the data they have uploaded is an important issue that needs to be resolved in permissionless blockchains. One other problem in permissionless blockchains is that data storage on the blockchain is very expensive. So there are many approaches that they store data on decentralized data storages like InterPlanetary File System (IPFS) [22].

Tracr™is an example of SCM that uses the public blockchain Ethereum. Tracr has been introduced by the De Beers Group to monitor diamonds from the mining stage, through cutting and polishing, and finally to the jewellers [23]. This system provides tamper-proof assurance of the diamond's source [24].

Musamih et al. [25] proposed blockchain-based system for pharmaceutical supply chain to address counterfeit drugs issue that is one of the consequences of complex healthcare supply chains structures. They use smart contract over Ethereum platform to define different functions of the different stakeholders (entities) in a supply chian. Such as initializing (manufacturing) a Food and Drug Administration (FDA) approved drug on blockchain. In their approach the manufacturer and distributors can update data captured by IoT devices on the blockchain. Their scheme provides the capability of uploading product images to the IPFS, where the IPFS hash of the image is published on Ethereum blockchain. The smart contract authenticate stakeholders using their Ethereum address.

Salah et al. [26] proposed a SCM solution based on the Ethereum Blockchain and IPFS to solve the problem of traceability in the agricultural supply chain where it is difficult to track and trace products in centralized controlled supply chain. Accordingly, in the event of contamination, identifying the source will be easier in blockchain based SCM solution. They employ Ethereum smart contracts to automate and enforce the rules and regulations of the supply chain and execute specific actions automatically when certain conditions are met. In their approach details of the product is captured and saved on IPFS as images. The details can be the time-stamped corp growth images. Hash of the stored file in IPFS is stored in the smart contract. The authentication of entities in their proposed approach is based on Ethereum address.

Toyoda et al. [27] proposed a novel product ownership management system (POMS). They have implemented POMS on a Ethereum smart contract. In their proposed approach, the manufacturer assign Electronic Product Code (EPC) to each product and write that into the RFID tag attached to the product. The product's EPC is registered on the smart contract. Furthermore, functions such as product owner transformation, incentivising entities to follow POMS protocol, and unauthorized party prevention are enabled by their smart contract. Toyoda et al. claimed PMOS system makes counterfeiters' efforts to clone real tags ineffective.

Many SCM schemes developed over permissionless blockchains, such as those in [25, 26, 27], do not preserve the privacy of their participants. Given the transparency of transactions on permissionless blockchains, these schemes could potentially compromise participants privacy. AlTway et al. presented Mesh [28], a supply chain solution over Ethereum smart contracts. Mesh uses group signatures to preserve the privacy of participant and employs a forward secrecy approach to preserve the confidentiality of data. Both are required to protect business secrets. However, Mesh requires a centralized Mesh server to keep the protocol going. Hence, we need to find a way to assure the data is authorized while it is stored off chain.

Mesh [28] uses Petersen's group signature [29] to authorize members of a supply chain. For each supply chain, all entities that are involved in that supply chain must build a group. The group must have a group manager which sends a list of identities of all group members to the Mesh Server and the Mesh's SupplyChain smart contract (C_{SC}) instance. For each smart contract, a separate instance of C_{SC} should be created. Only the group members of a supply chain are able to upload data to the related C_{SC}. The data is stored on Ethereum's Blockchain. To limit the upload access to the group members, Altawy et al. [28] use Petersen's group signature [29] for members' authorization. Consequently, C_{SC} will be able to verify the membership of an entity without learning the identity of that user. However, it is clear that the transaction is from a known group without knowing which group member is sending the data. Also, Petersen's scheme provides plausible anonymity which means that although group members are locally anonymous, their identity can be revoked by the manager, Mesh Server or coalition of members.

There are some shortcomings in Mesh. First, it is only locally anonymous. Second, The length of each signature is linear in group size. So, for big groups, it will not be applicable to store signatures on public Blockchains such as Ethereum. Third, groups in Mesh are static. The managers should submit entities' IDs before they start using the C_{SC}. Therefore, it does not allow dividing a group into two sub-groups or merging two groups into one bigger group.

B. Blockchain Based Version Control Systems

The rise of blockchain technology has opened up new possibilities for decentralized VCSs. By leveraging the decentralized and trustless nature of blockchain, VCSs over blockchain can provide several benefits, such as increased transparency, immutability, and resistance to censorship. One example of a VCS over blockchain is Gitcoin [30], which combines the Git VCS with the Ethereum blockchain. Gitcoin allows developers to receive rewards for contributing to open-source projects and offers a transparent and decentralized funding platform for developers. Blockchain based VCS solutions use public-key to build the identity of a user which does not preserve the anonymity of developers. Moreover, these approaches usually do not support private repositories [31].

C. Privacy in Blockchain

Bitcoin [32] is the first blockchain platform provides an authenticated tamper-proof record of transactions maintained in a peer-to-peer network. Transactions on Bitcoin are linked together. Many research works such as [33, 34, 35] showed that analyzing the transaction graphs, values, and dates of transactions helps to retrieve information that leads to attribute ownership of Bitcoin addresses. Consequently, the identity of the owners could be revealed easier than was expected. Researchers have proposed privacy-focused solutions, such as CoinJoin [36], in response to privacy concerns. However, these solutions do not necessarily provide complete privacy guarantees.

Zerocash [37] is a cryptocurrency that shares similarities with Bitcoin in terms of its blockchain technology. However, Zerocash uses zero-knowledge proofs to break the link between the input of a new transaction and the Unspent Transaction Output (UTXO) of an existing transaction on the blockchain, thereby providing enhanced privacy. Zerocash can be implemented on top of Bitcoin or other blockchains, and the use of private transactions is optional.

With the possibility of implementing the idea of smart contracts on Blockchain platforms (such as Ethereum), we can see an increasing growth in the implementation of various financial Decentralized Applications (DApps) on Blockchain. Transferring financial activities to the blockchain platforms without paying attention to the privacy of users can have worrying consequences. Because of this, many efforts have been made to provide methods to protect people's privacy in the blockchain. Such as, Zeestar [38], Zkay [39], Hawk [40], etc. These methods are general approaches that aim to preserve the privacy of users.

Hawk [40] is a smart contract compiler that enables users to interact with a smart contract in a secure manner. Hawk developers implemented zkSNARKs protocols on Ethereum smart contracts, inspired by coin minting and pouring from Zerocash. This allows the use of native cryptocurrency on smart contracts anonymously by hiding money transfers within a smart contract. However, this approach is not suitable for many DApps. For example, where users need to transfer other tokens on Ethereum anonymously, or they want to employ functionalities of smart contracts other than token transformations (Pour). Additionally, Hawk requires a protocol manager.

Narula et. al [41] proposed zkLedger which is a private but also audiable transaction ledger. Their focus is on banks that want their transactions be encrypted; but, let regulators have an insight into bank assets and trades. Auditor can query the bank and have a verifiable proof that the answer actually is true. zkLedger leverages Peterson commitment [29] for committing values. Peterson commitment enables linear functions over transaction values. For example, ratios, percentages, sums, averages, etc. zkLedger leverages zero-knowledge proofs to ensure that banks have valid transactions, i.e., they had consent to transfer the asset, they had enough asset to transfer, and the assets are neither created or destroyed.

Bowe et. al presented Zexe [42] that addresses privacy and scalability in Ethereum blockchain. They propose decentralized private computation (DPC) scheme which extends Zerocash [37]. Zexe leverages DPC to enable offline computation. The offilie computations produces publicly-verifiable transactions that prove correctness of these offline executions. However, it requires cryptographic expertise for implementing new applications [38] and a separate trusted setup for each application[43]. Xiong et al. [43] proposed a new DPC scheme in VeriZexe which needs only one single universal setup to be able to support any number of applications.

ZeeStar [38] is a compiler that converts a smart contract into an anonymous smart contract where computations are performed off the blockchain on the user side. The encrypted values are then assigned to variables on the smart contract, ensuring that no one knows the actual values of the variables. To verify the accuracy of the computation done by the user, a zero-knowledge proof is uploaded along with the newly encrypted variables. This means that the new state of the smart contract is computed off chain and its correctness is verified by nodes on the Ethereum platform. To achieve this capability, the user must rewrite the smart contract and add privacy annotations to the code. Accordingly, the user specifies which parts of the code should be private and which parts can be public.

ZeeStar [38] provides more applicability than Hawk [40]. However, it still needs to verify the transaction sender according to its address to determine if the sender has permission to execute the rest of the called function. Therefore, using address-based authentication to verify the transaction sender of a smart contract can lead to significant information leakage.

IV. Outline of the Framework

In this section, we introduce a framework for a verifiable authentication mechanism. This framework aims to maintain a shared DAG that ensures the anonymity of data uploaders and prevents linking data ownership transfers. Additionally, participants should be able to authenticate the uploaded data. This solution is suitable for various applications, including SCM and VCS.

The necessity of a trustless and managerless scheme has prompts us to utilize public blockchain platforms. As a result of using public blockchain platforms which supports Turing complete smart contracts (such as Ethereum [44]), each DAG maintained by our solution is not restricted to a particular group of entities, nor it is managed by a group of managers who must be trusted to adhere to their commitments. The use of a public blockchain platform offers a flexible environment where, at the most basic level, two chains from different DAGs can choose to merge or a chain can be forked to create two subchains. In addition, at higher levels, smart contracts can incentivize or enforce adherence to certain rules by entities. This freedom of action facilitates communication between different supply chains worldwide without requiring complex bureaucratic procedures. Also, it provides all the functionalities of VCSs, including branching, merging, and so on.

We aim to preserve the privacy of the data uploaders and the process in transferring ownerships. Therefore, while verifying the authenticity of the uploaded data, we aim to preserve the privacy of the data uploaders. To accomplish this goal, we utilize zero-knowledge proof protocols to propose an anonymous authentication approach that is suitable for this scheme. The framework is described bellow:

A. Components of the framework

1) **Entities/Auditors**: Entities represent the participants within the framework. In a food supply chain context, these entities can be farmers, transportation companies, packaging manufacturers, and retailers. They have the option to mint an authentication token by *initiating* a DAG or to receive such a token from a preceding entity. Auditors, too, fall under the category of entities. Their primary role is to verify the authenticity of uploaded data. For instance, when an entity receives a product from its preceding entity, it may audit the product's entire history.
2) **Zero-Knowledge Proof**:In the framework, zero-knowledge proofs authenticate the ownership of tokens while preserving anonymity and are essential for verifying off-chain data records. They also play a role in token transfer, merging, and division while preserving unlinkability. Entities utilizing this framework for SCM or VCS generate these proofs. For a detailed background on zero-knowledge proof systems, refer to Appendix A.
3) **Permissionless Blockchain Platform**: The permissionless blockchain platform offers a decentralized and irreversible system that securely manages the commitments of authentication tokens. It not only provides the capability to design smart contracts specifically for verifying zero-knowledge proofs, ensuring a trustless environment, but also maintains and updates its states.
4) **Smart Contract**: In the framework, a smart contract commits and transfers the ownership of authentication tokens, ensuring anonymity in a trustless, secure environment. These tokens are crucial for the anonymous authentication of data uploaders. They are committed to a Merkle tree maintained by the smart contract. The smart contract also handles the verification of zero-knowledge proofs for operations like transferring, merging, and division. Since it's secured by the blockchain, there's no need for a trusted third-party for verification. Any invalid proofs, possibly crafted by adversaries, are rejected by the smart contract. Thus, only legitimate proofs from authorized entities can alter the protocol's state.
5) **Decentralized Cloud Storage (DCS)**: The framework stores the uploaded data on an affordable, trustless and managerless data storage solution. It leverages decentralized cloud storage services such as Filecoin [45]. Filecoin is a transparent, decentralized marketplace that utilizes blockchain technology for storing and retrieving data. In Filecoin, storage providers contribute storage capacity to the network and receive Filecoin cryptocurrency (FIL) in exchange. In this framework, our main focus will be on the privacy and unlikability of tokens which is conducted on the premissionless blockchain platform (e.g., Ethereum).

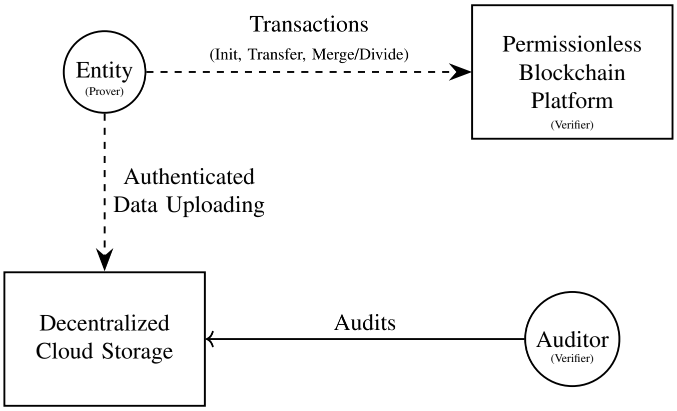
Transactions
(Init, Transfer, Merge/Divide)
Entity
(Prover)
Permissionless Blockchain Platform
(Verifier)
Authenticated Data Uploading
Decentralized Cloud Storage
Audits
Auditor
(Verifier)

Fig. 1. Entities participate in maintaining DAG-structured data records. They manage anonymous authentication tokens by sending transactions to the smart contract on blockchain platform. These tokens also enable them to upload authenticated data to decentralized cloud storage. Auditors ensure the authenticity of this uploaded data.

B. The framework functions

1) **Init**: The initiator of a DAG commits a transaction to the blockchain, embedding an authentication token that includes a public key, to initialize the DAG.
2) **Transfer**: The owner of an authentication token transfers the ownership to a new entity with a new public key using zero-knowledge proofs to hide the link between two owners.
3) **Merge/Divide**: The owner of two authentication tokens merges the commitments to a new commitment. Or, the owner of a authentication token divides the authentication token to two new commitments.
4) **Authenticated Data Uploading**: The data uploader uploads the actual data on a decentralized off-chain data

storage. The data includes the zero-knowledge proof of owning a valid authentication token whose commitment is on blockchain.

5) **Audit**: The auditors verify the proof of authenticity of the uploaded data.

Furthermore, the framework employs encryption schemes with perfect forward secrecy to encrypt sequenced data in a chain, which ensures that the confidentiality of exchanged keys from earlier sessions is not compromised even if the long-term secret keying material is disclosed [46]. As a result, when using this solution in a supply chain for food, any entity can decrypt and view the history of the food up until the point of transferring ownership to the next entity, but cannot access the subsequent records [28]. Similarly, in a version control system, a developer can access the entire history of a private repository.

C. Threat model

The primary goals an adversary might pursue within the framework are anticipated to be: (1) compromising anonymity by identifying the uploader of a data record; (2) undermining unlinkability by establishing connections between authentication tokens related to a DAG; and (3) violating data authenticity by uploading an unauthenticated data record to a DAG.

First, In this framework, data records are stored in DCS, while the management of authentication tokens is maintained on the blockchain platform. Entities can provide a zero-knowledge proof of owning an authentication token while uploading a data records. Due to the zero-knowledge property of the proof (Appendix A), any polynomial time adversary (PPT) is unable to compromise their anonymity. Second, The framework enables the entities to provide a zero-knowledge proof of owning an authentication token and transferring it without revealing which authentication token the entity owned. Finally, the soundness property of zero-knowledge proof schemes, as detailed in Appendix A, ensures that no polynomial-time adversary (PPT) can forge a zero-knowledge proof of owning an authentication token if they do not actually possess one.

V. Discussion

The DAG structure of data sequences in this framework makes it a perfect fit for SCM and VCS applications. The protocol maintains data integrity and public verifiability of data records while preserving the privacy of the data uploaders and confidentiality of data. The framework provides an anonymous and unlinkable authentication protocol for off-chain data storage authentication. Methods like Mesh [28], which store data on the Blockchain, can ensure local anonymity and authenticity of the uploaded data; However, they are expensive sinece they must store data on the blockchain plantform. For instance, based on the current ETH/USD rate, storing 1 MB of data on the blockchain would cost $44,118 USD[1]. While, off-chain data storage cuts the storage cost; therefore, makes the protocol practical. Other methods that utilize decentralized cloud storage, such as [25] and [26], do not propose any approach for authenticating the uploaded off-chain data.

The framework's efficiency, integrity, anonymity and unlinkability, makes it feasible for small businesses to utilize it in their own SCM. An example of such businesses could be online retailers on social media platforms like Instagram, which might be operated by a single individual. These businesses can allow their end customers to see the entire product history. While, the framework ensures the unlinkability of its participants. This enables businesses to participate in a supply chain without fear of revealing their business secrets through a trustless foundation

Moreover, for the the VCS applications, the framework enables developers to participate in collaborative software development by using it as a VCS. It offers essential features like forking, branching, and merging that are necessary for VCSs. Therefore, developers can work together on software development projects while keeping their identities private. It's worth mentioning that VCS applications primarily aim to track changes made to a document. These differences, at the application level, can be extracted from the data record layer. Investigating on the application level is out of the scope of the proposed framework outline.

VI. Conclusion

In this paper we have proposed the outline of a solution that enable entities in a supply chain, or developers working on a same code base to upload their data records anonymously. The scheme enables the data auditors that can be the end costumers of a supply chain to access the history of a product, verify the authenticity of the data; while, preserving the privacy of the data uploader. The solution is inexpensive. Hence, this property makes it practical for even very small businesses to employ this solution for their supply chain managements.

Multiple supply chains and version control systems can use one instance of the smart contract without needing to trust the smart contract deployer. Furthermore, the protocol provides the unlinkability in tokens transferring, merging, and division. Such that an adversary cannot distinguish anonymous authentication tokens that are related to each other in a supply chain or version control system. The framework contains algorithms for each of processes on anonymous authentication tokens, gas efficient Merkle tree update and auditing the uploaded data. We also provide a security analysis to justify the claimed security properties of the protocol. We will present a concrete solution accompanied with an implementation in a separate paper.

VII. Acknowledge

We would like to appreciate the insightful comments and constructive feedbacks of the reviewers which contributed to the refinement and improvement of the paper.

[1] Ethereum price is $1,870.00 USD and the gas price is 36.00 Gwei at the time of writing this paper

References

[1] F. Tian, "An agri-food supply chain traceability system for china based on rfid & blockchain technology," in *2016 13th International Conference on Service Systems and Service Management (ICSSSM)*, pp. 1–6, 2016.

[2] W. C. Tan and M. S. Sidhu, "Review of rfid and iot integration in supply chain management," *Operations Research Perspectives*, vol. 9, p. 100229, 2022.

[3] S. Safaric and K. Malaric, "Zigbee wireless standard," in *Proceedings ELMAR 2006*, pp. 259–262, 2006.

[4] S. Bluetooth, "Specification of the bluetooth system-covered core package version: 4.0," *Bluetooth Special Interest Group*, 2010.

[5] S. Aich, S. Chakraborty, M. Sain, H.-i. Lee, and H.-C. Kim, "A review on benefits of iot integrated blockchain based supply chain management implementations across different sectors with case study," in *2019 21st International Conference on Advanced Communication Technology (ICACT)*, pp. 138–141, 2019.

[6] Y. Bi, K. Fan, K. Zhang, Y. Bai, H. Li, and Y. Yang, "A secure and efficient two-party protocol enabling ownership transfer of rfid objects," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[7] V. Cherneva and J. L. Trahan, "Tp-otp: Two-party, ownership transfer protocol for rfid tags based on quadratic residues," in *2021 IEEE Green Energy and Smart Systems Conference (IGESSC)*, pp. 1–6, 2021.

[8] X. Yang, C. Xu, C. Li, *et al.*, "A privacy model for rfid tag ownership transfer," *Security and Communication Networks*, vol. 2017, 2017.

[9] "Git scm," 2023.

[10] "Mercurial scm," 2023.

[11] Y. Yang, S. Pan, and E. Ballot, "Mitigating supply chain disruptions through interconnected logistics services in the physical internet," *International Journal of Production Research*, vol. 55, no. 14, pp. 3970–3983, 2017.

[12] M. S. Sodhi and C. S. Tang, "Supply chain management for extreme conditions: Research opportunities," *Journal of Supply Chain Management*, vol. 57, no. 1, pp. 7–16, 2021.

[13] "Ibm supply chain intelligence suite - solution - canada — ibm," 2023.

[14] S. Saberi, M. Kouhizadeh, J. Sarkis, and L. Shen, "Blockchain technology and its relationships to sustainable supply chain management," *International Journal of Production Research*, vol. 57, no. 7, pp. 2117–2135, 2019.

[15] N. Kshetri, "1 blockchain's roles in meeting key supply chain management objectives," *International Journal of Information Management*, vol. 39, pp. 80–89, 2018.

[16] M. Nadler and F. Schär, "Tornado cash and blockchain privacy: A primer for economists and policymakers," *Available at SSRN 4352337*, 2023.

[17] "U.s. treasury sanctions notorious virtual currency mixer tornado cash — u.s. department of the treasury."

[18] "Iran to execute alleged porn site developer - neowin."

[19] "Hyperledger fabric – hyperledger foundation."

[20] P. Helo and Y. Hao, "Blockchains in operations and supply chains: A model and reference implementation," *Computers & Industrial Engineering*, vol. 136, pp. 242–251, 2019.

[21] "Walmart case study – hyperledger foundation."

[22] "Ipfs powers the distributed web," 2023.

[23] N. Kshetri, "Blockchain systems and ethical sourcing in the mineral and metal industry: a multiple case study," *International Journal of Logistics Management*, vol. 33, pp. 1–27, 2 2022.

[24] "Tracr – de beers group."

[25] A. Musamih, K. Salah, R. Jayaraman, J. Arshad, M. Debe, Y. Al-Hammadi, and S. Ellahham, "A blockchain-based approach for drug traceability in healthcare supply chain," *IEEE Access*, vol. 9, pp. 9728–9743, 2021.

[26] K. Salah, N. Nizamuddin, R. Jayaraman, and M. Omar, "Blockchain-based soybean traceability in agricultural supply chain," *IEEE Access*, vol. 7, pp. 73295–73305, 2019.

[27] K. Toyoda, P. T. Mathiopoulos, I. Sasase, and T. Ohtsuki, "A novel blockchain-based product ownership management system (poms) for anti-counterfeits in the post supply chain," *IEEE Access*, vol. 5, pp. 17465–17477, 2017.

[28] R. AlTawy and G. Gong, "Mesh: A supply chain solution with locally private blockchain transactions," *Proceedings on Privacy Enhancing Technologies*, vol. 3, pp. 149–169, 2019.

[29] H. Petersen, "How to convert any digital signature scheme into a group signature scheme," in *International Workshop on Security Protocols*, pp. 177–190, Springer, 1997.

[30] M. Król, S. Reñé, O. Ascigil, and I. Psaras, "Chainsoft: collaborative software development using smart contracts," in *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, pp. 1–6, 2018.

[31] "Radicle: Sovereign code infrastructure," 2023.

[32] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized business review*, p. 21260, 2008.

[33] F. Reid and M. Harrigan, *An Analysis of Anonymity in the Bitcoin System*, pp. 197–223. New York, NY: Springer New York, 2013.

[34] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *Financial Cryptography and Data Security* (A.-R. Sadeghi, ed.), (Berlin, Heidelberg), pp. 6–24, Springer Berlin Heidelberg, 2013.

[35] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: Characterizing payments among men with no names," *Commun. ACM*, vol. 59, p. 86–93, mar 2016.

[36] G. Maxwell, "Coinjoin: Bitcoin privacy for the real world," 2013.

[37] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin (extended version)," 2014.

[38] S. Steffen, B. Bichsel, R. Baumgartner, and M. Vechev, "Zeestar: Private smart contracts by homomorphic encryption and zero-knowledge proofs," in *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 179–197, May 2022.

[39] S. Steffen, B. Bichsel, M. Gersbach, N. Melchior, P. Tsankov, and M. Vechev, "Zkay: Specifying and enforcing data privacy in smart contracts," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, (New York, NY, USA), p. 1759–1776, Association for Computing Machinery, 2019.

[40] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 839–858, May 2016.

[41] N. Narula, W. Vasquez, and M. Virza, "zkLedger: Privacy-Preserving auditing for distributed ledgers," in *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, (Renton, WA), pp. 65–80, USENIX Association, Apr. 2018.

[42] S. Bowe, A. Chiesa, M. Green, I. Miers, P. Mishra, and H. Wu, "Zexe: Enabling decentralized private computation," in *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 947–964, 2020.

[43] A. L. Xiong, B. Chen, Z. Zhang, B. Bünz, B. Fisch, F. Krell, and P. Camacho, "Veri-zexe: Decentralized private computation with universal setup," in *32st USENIX Security Symposium (USENIX Security 23)*, 2023.

[44] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform [white paper]," 2014.

[45] "A decentralized storage network for humanity's most important information — filecoin," 2023.

[46] L. Chen and G. Gong, *Communication system security*. CRC press, 2012.

[47] E. Ben Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*, pp. 459–474, 2014.

[48] J. Bonneau, I. Meckler, V. Rao, and E. Shapiro, "Mina: Decentralized cryptocurrency at scale," 3 2020.

[49] S. Goldwasser, S. Micali, and C. Rackoff, "Knowledge complexity of interactive proof-systems.," *Conference Proceedings of the Annual ACM Symposium on Theory of Computing*, pp. 291–304, 1985.

[50] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Advances in Cryptology — CRYPTO' 86* (A. M. Odlyzko, ed.), (Berlin, Heidelberg), pp. 186–194, Springer Berlin Heidelberg, 1987.

[51] A. Nitulescu, "Lattice-based zero-knowledge snargs for arithmetic circuits," in *Progress in Cryptology – LATINCRYPT 2019* (P. Schwabe and N. Thériault, eds.), (Cham), pp. 217–236, Springer International Publishing, 2019.

[52] G. Brassard, D. Chaum, and C. Crépeau, "Minimum disclosure proofs of knowledge," *Journal of Computer and System Sciences*, vol. 37, no. 2, pp. 156–189, 1988.

[53] J. Kilian, "A note on efficient zero-knowledge proofs and arguments (extended abstract)," in *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '92, (New York, NY, USA), p. 723–732, Association for Computing Machinery, 1992.

[54] N. Bitansky, R. Canetti, A. Chiesa, and E. Tromer, "From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, (New York, NY, USA), p. 326–349, Association for Computing Machinery, 2012.

[55] "Zcash genesis block - youtube," 2016.

[56] "Github - aztecprotocol/ignition-verification: Repository to verify contributions to the aztec ignition ceremony," 2020.

[57] J. Groth, "On the size of pairing-based non-interactive arguments," *Cryptology ePrint Archive*, 2016.

[58] R. Gennaro, C. Gentry, B. Parno, and M. Raykova, "Quadratic span programs and succinct nizks without pcps," in *Advances in Cryptology – EUROCRYPT 2013* (T. Johansson and P. Q. Nguyen, eds.), (Berlin, Heidelberg), pp. 626–645, Springer Berlin Heidelberg, 2013.

[59] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev, "Scalable, transparent, and post-quantum secure computational integrity." Cryptology ePrint Archive, Paper 2018/046, 2018.

[60] S. Ames, C. Hazay, Y. Ishai, and M. Venkitasubramaniam, "Ligero: Lightweight sublinear arguments without a trusted setup," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, (New York, NY, USA), p. 2087–2104, Association for Computing Machinery, 2017.

[61] E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward, "Aurora: Transparent succinct arguments for r1cs," in *Advances in Cryptology – EUROCRYPT 2019* (Y. Ishai and V. Rijmen, eds.), (Cham), pp. 103–128, Springer International Publishing, 2019.

[62] S. Setty, "Spartan: Efficient and general-purpose zksnarks without trusted setup." Cryptology ePrint Archive, Paper 2019/550, 2019.

[63] S. Fu and G. Gong, "Polaris: Transparent succinct zero-knowledge arguments for r1cs with efficient verifier," *Proceedings on Privacy Enhancing Technologies*, vol. 2022, pp. 544–564, 2022.

Appendix A
Preliminaries: zero-knowledge proofs

In this section we explain zero-knowledge proof protocols as one of the main components of the solution. Zero-knowledge proof is one of the most useful tools employed to preserve the privacy of users in blockchain platforms. There are many existing blockchain-based projects using zero-knowledge proofs such as Zerocash [47], and Mina [48]. Zero-knowledge proofs are first introduced by Goldwasser, Micali, and Rackoff [49]. They are privacy-enhancing cryptographic techniques that enable one prover to convince a verifier that a given NP statement is true without disclosing any additional information. A zero-knowledge proof must satisfy these properties:

1) **Completeness**: If a statement is true, the prover can convince the verifier if given the corresponding witness.
2) **Soundness**: If a statement is false, it would be impossible for a malicious prover to fool the verifier to accept the false statement as a true statement.
3) **Zero-knowledge**: The proof reveals nothing but the validity of the statement. Particularly it does not reveal the witness of the prover.

Most zero-knowledge proofs need interactions between the prover and the verifier. Non-interactive zero-knowledge proof (NIZK) are a variant of zero-knowledge proofs that do not need these interactions. We can apply Fiat-Shamir [50] transformation heuristic to transform interactive zero-knowledge protocols to NIZK proofs.

In statistically sound proof systems, the communications might be as much the size of the witness [51]. However, we can settle for computationally sound proof systems which are sound against computationally bounded provers. These systems are called Argument Systems [52]. Kilian showed that if we assume Collision-Resistant Hashing (CRH) functions existed, we can have succinct argument systems in which communication can be cut down and make the verifier run much faster than it would in previous NP verifications [53]. By generating a common reference string (CRS) a head of protocol execution and applying Fiat-Shamir transformation, we can have a Succinct Non-Interactive Argument (SNARG) proving system.

In a simple language, a prover in a SNARG proving system proves that there exists a witness for an NP statement (the statement is true). However, it is not enough in many applications. Usually, she must prove that she knows that witness too (proof of knowledge). The SNARGs in which a prover proves that she knows the witness are called Succinct Non-Interactive Argument of Knowledge (SNARK) [54]. Bitansky et al. proved that if and only if Extractable Collision-Resistant Hash (ECRH) functions exist, verifier of SNARKs will exist [54]. Hence we will have computational knowledge soundness in SNARK instead of computationally soundness that was in SNARG. SNARKs can be modified so that a verifier does not learn anything about the witness then we will have zkSNARKs [54].

There are two approaches for design zkSNARK schemes in terms of the set-up. One is the trusted setup and the other, transparent set-up. In trusted set-up, we need to rely on a trusted third party to run a probabilistic algorithm to generate secret randomness; then, build and publish the CRS. The secret randomness must be generated via a multi-party computation (MPC) then the generator(s) must be trusted to remove the secret randomness in the end. This will happen usually in a ceremony like the one Zerocash [37] had for generating the secret randomness and then destroying it in a live YouTube video stream [55], or the AZTEC CRS MPC setup [56]. The proposed implementations of zkSNARKs by Groth [57] and Gennaro et al. [58] require a trusted setup phase, during which a common reference string (CRS) is generated, and the proving and verification keys are extracted from it. For transparent set-up, such as Stark [59], Ligero [60], Aurora [61], Spartan [62] and Polaris [63] in the random oracle case, to just list a few, their proof sizes and verifiers' cost are much higher than the cases with trusted set-up.

The efficiency of different approaches of zero-knowledge proving system implementations can be compared in different aspects, which makes each of them suitable for different applications. These aspects are proving computational cost (prover complexity), The number of transactions between the prover and the verifier (round complexity), communication size between the prover and the verifier (proof length), verifying computational cost (verifier complexity), storage and computational cost of generating public parameters, such as CRS (setup cost) in trusted set-up.