Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Trusted-auditing chain: A security blockchain prototype used in agriculture traceability

Moyixi Lei [a,b,c], Shuangyin Liu [b], Na Luo [a,c], Xinting Yang [a,c], Chuanheng Sun [a,c,*]

[a] National Engineering Research Center for Information Technology in Agriculture, Beijing 100097, China
[b] College of Information Science and Technology, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China
[c] National Engineering Laboratory for Agri-product Quality Traceability, Beijing 100097, China

ARTICLE INFO

Keywords:
Agricultural traceability
Blockchain
Security auditing
Privacy-preserving
Non-interactive zero-knowledge proof

ABSTRACT

Traceability systems have changed the way food safety is managed and data is stored. Blockchain tracking services now provide customers with an infrastructure that allows them to easily access data online. However, there are limitations to these new capabilities, such as a lack of transparency and the existence of privacy and security challenges. Additionally, as the need for more agile, private, and traceability secure data solutions continues to grow exponentially, rethinking the current structure of blockchain agricultural traceability is mission-critical for a country. By leveraging and building upon blockchain's unique attributes, including tamper-evident, security hash crypto-data, and distributed ledger, we have proposed a prototype that allows traceability data to be reliably stored via blockchain while simultaneously being secured, with completeness auditing to enhance credibility. The result, the trusted auditing chain (TA chain), is a flexible solution that assures data security and solves challenges such as scalability and privacy-preserving. The TA chain works through Schnorr-style non-interactive Zero-knowledge proof to support security automatical choose privacy augmented. In addition, The TA chain can audit more than 1000 transactions within 1ms, and its error stabilizes below the 250 μs, which proves a security and fair traceability system to assure that data is distributed and reliably, and provably audited.

1. Introduction

Food security has attracted lots of authorities and enterprises concerned (Zhu et al., 2022). They come up with a traceability method to protect the agricultural foods that can be stored, processed, and sold timely, and also ensure each item can be tracked in the world, but because of the widely distributed traceability information, the privacy within the process of the traceability cannot be sure. For example, food traceability systems without privacy-preserving technology are vulnerable to malicious attacks, and incorrect information can lead not only to financial losses but also to health hazards (Lei et al., 2022). Traceability networks often involve many parties and sites, as well as a large flow of information and private data. With the involvement of more nodes, more exchanges, and supervision being conducted far from authority institutions, if not checked in time will come the opportunity for error, whether it is unintentional or malicious. For example, the malicious *Supplier* can store illegal materials or threatening information on agricultural food labels (Dodd and Solutions, 2022).

Therefore, just in response to the above phenomenon, some researchers proposed the use of blockchain technology for the storage of food traceability data (Wang et al., 2022a). In the Bonde Thylstrup et al. (2022), the author was concerned about the concept of accountability, explainability, and speculation, which disclose to the importance of traceability. That is why it needs the Blockchain to explore its traceability potential. Blockchain is a distributed system for storing important information such as account transaction records, electronic contracts, and invoice documents (Liu et al., 2022b). Compared with traditional traceability systems, in blockchain, the data can be protected on-chain, generally, is hashed transactions on the distributed ledger (Xie et al., 2022). The application of cryptography allows transactions on the chain to be stored in abstract form, which avoids the drawbacks of explicit access to data (Liu et al., 2022a). For example, a survey of the fish farm platform displays a whole traceability ecosystem with diverse information from sensors, operators, entertainment, and so on (Hang et al., 2020).

But this type of ledger without security verification will be unfavorable for humans. Then for a greatly im-proved food security and

* Corresponding author.
E-mail address: sunch@nercita.org.cn (C. Sun).

https://doi.org/10.1016/j.heliyon.2022.e11477
Received 15 June 2022; Received in revised form 4 September 2022; Accepted 2 November 2022

2405-8440/© 2022 Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

utility traceability system, auditing can be introduced into the blockchain (Dasaklis, 2020) introduces a trusted third party to verify information of transactions, but it will expose the transaction process to the third party, besides the much more sensitive information, which privacy will uncover, and around that traceability system things will be exposed to adversaries. But the auditing-related process can be improved to prevent privacy from the opponents (Wang et al., 2019c; Francati et al., 2021), which proves off-chain storage in an auditing framework is feasible. However, the former stops opponents from attacking or tampering with traceability, but without completeness transactions auditing still leaves bugs to malicious nodes. For full auditing, the latter pronounces an auditing way with off-chain combined, so it has to download all transactions and cannot be audited in real-time.

Alliance cooperation between manufacturers and their *Suppliers* should be aligned with the goal of sustainable development. There is an increasing demand for food safety in the current competitive market and industry environment (Khanfar et al., 2021). Because blockchain can maintain a track record of transactions, it increases the ability to audit information and reduces the cost of guarantees and the time spent on the audit process (Perera and Abeygunasekera, 2022). However, a lack of security auditing or non-completeness auditing will lead to challenges to privacy leakage, data security storage, and data effective sharing (Chenli et al., 2022; Brebu et al., 2022). Therefore, making blockchain can be safely traced is a great way to link *Suppliers* and customers, introducing an *Auditor* into the traceability system will enhance the credibility of traceability results. For the *Auditor*, verifying the traceability information including private information but without using it directly, will be better for transaction verification and food security.

Hence, in this paper, inspired by blockchain, we create a trusted audit chain (TA chain), which needs to be permissioned. Furthermore, based on this, we propose a Schorr-based non-interactive zero-knowledge proof-based approach for security auditing to support non-interactive evidence generation and verification throughout the audit period. Meanwhile, audit cache sequences as an important way for online auditing to be used, which can significantly improve the efficiency of online auditing and create a gap with offline auditing. Simulation experiments compare the difference between online and offline auditing, considering the relationship between audit time, audit standard deviation error, system throughput, and the number of audited transactions when using TA chains for a single *Supplier* and multiple *Suppliers*. This paper references use cases of fisheries germplasm resources, where both public and private data are used for valuable fisheries resource data, to test the feasibility of our model.

The contributions of this paper are as follows:

- This study is designed to be set up in a permissioned blockchain to computing on privacy data directly, using a privacy-preserving style distributed ledger to build a trace-ability network where internal transactions have already been fully trusted by each other.
- Using audit sequences for each transaction when it requires to be stored in the distributed ledger, the *Auditor* gets fast and completes auditing effectively online and offline, this decentralized and fair auditing also gives much security.
- This work proposed a Non-interactive Zero-knowledge proof to solve the combination with the transactions and *Auditor* to finish the auditing. Moreover, to protect the privacy of information, we proposed an approach to increase ledger transaction security by using Schorr.

The rest of this paper is organized as follows. Section 2 presents previous works and makes a comparison of each of them. Section 3 explains the schema and methodology in detail. Section 4 implements the prototype and makes an evaluation. Finally, Section 5 concludes with details demonstrating our study's contribution clearly and proposes some feasible future directions.

2. Literature review

The concept of data integrity auditing of remote servers was first introduced by (Ateniese et al., 2007), where data owners can periodically check the integrity of the data they provide on remote servers. Liu et al. (2014) proposed a conformance service auditing scheme to verify whether the data cloud provides the promised consistency. Jin et al. (2018) proposed a public auditing scheme with support for dynamic data to enable efficient handling of real-time data. Wang et al. (2019b) proposed an identity-based public provable data possession scheme to eliminate the complex certificate management that exists in public key infrastructures. In (Wang et al.), the authors considered an identity-based online/offline security auditing scheme that can reduce the online computational cost of data owners by precomputing metadata offline. Also, Rabaninejad et al. (2020) proposed a similar offline idea. Offline auditing can reduce operational costs, which was an efficient payback to the reviewer.

However, both concentrate on offline auditing of metadata and do not consider the issues of communication and security in offline auditing. Later, researchers introduced blockchain, which is tamper-proof, so they attempted to integrate blockchain into remote auditing schemes, trying to use blockchain to check the security of data in remote storage (Xue et al., 2019). Bitcoin (Nakamoto, 2008) is the first-come blockchain to ensure financial safety. Ethereum (Dannen, 2017) can also guarantee on a remote server, but not like Bitcoin, Ethereum is much more flexible to control the transfer of various coins and the storage is gradually growing.

Wang et al. (2019a) used the blockchain to build a decentralized auditing framework whose responsibility for auditing is taken care of by its nodes. This solution solves the problem of trusting third parties, but the audit nodes are included in the attack considering security issues. Konkin and Zapechnikov (2021) used zero-knowledge proofs to harden data security, but do not take into account the computational process of data interaction. Decentralized and fair auditing features a significant point of a distribution auditing condition. System Blockstore fulfilled fairness because file owners are responsible for selecting storage nodes and challenge rates previously but authorities a third party to audit its data (Ruj et al., 2018). In Francati et al. (2021), authors designed decentralized and fair auditing, especially in the distributed storage system, but there is no privacy-preserving. For example, when a malicious auditor checks the transactions, they cannot ensure the privacy information will not be threatened. Yu et al. (2022) designed an offline scheme that eliminates communication between the vehicle and the cloud. This is a lightweight approach. Automatic auditing is also supported through the use of smart contracts.

In summary, there are many excellent blockchain-based auditing solutions available, such as Rabaninejad et al. (2020), Francati et al. (2021) and Yu et al. (2022), but most of them focus on offline auditing because auditing data in massive cloud storage requires communication issues to be considered. There-fore, a lightweight approach should be investigated. This work proposes an online cache audit that uses non-interactive zero-knowledge proofs to secure data, and the online audit method is more effective and stable than offline auditing. Based on such properties, we made a simple comparison between the TA chain and the existing audit system (Table 1), allowing our work to be presented more clearly.

3. Trusted auditing chain: our blockchain

3.1. Permissioned blockchain

The issue in creating the TA chain is to practically support complete, confidential auditing—an *Auditor* would not allow in-sight into single traceability transactions, but a *Supplier* should not be able to hide information from the *Auditor* during an audit, and the *Auditor* should be able to detect a deceived answer. Figure 1 depicts an overview of the TA chain, which contains five layers. The lower four layers are the main functions of the system, and the arrows between each layer represent the flow of processing data. The top layer is the auditing operation of the

Table 1. The comparison summary between the TA chain and current auditable systems (✓ means completely satisfied, ✔ means partly satisfied and (× means not satisfied.).

Authors and year	privacy-preserving blockchain	Decentralized and fair auditing	Security and completeness auditing	Auditing way
Liu et al. (2014)	×	×	✔	Online & offline
Jin et al. (2018)	×	✔	✓	Public
Wang et al. (2017)	×	✔	✔	Online & offline
Rabaninejad et al. (2020)	×	×	✔	Online & offline
Xue et al. (2019)	✔	✔	✔	Public
Wang et al. (2019a)	✓	×	✔	Public
Ruj et al. (2018)	✔	✔	✔	–
Francati et al. (2021)	✔	✓	✔	Offline
Yu et al. (2022)	✔	✔	✔	Offline
Trusted auditing chain, 2022	✓	✓	✓	Online

auditor, with the arrows representing the acquisition of data. Therefore, each of the functions mentioned in the overview is necessary. However, the ‘technical support’ layer is the main emphasis of the research program in this study and will be analyzed in detail in the article.

There are two major nodes in Ledger A, block-creators and auditors. Block-creators submit the transactions to an append-only Ledger A, which simply proposes all feasible transactions, such as the producers, the processors, the storage, the sellers, and so on, so they are *Supplier* nodes. The Ledger A maintains by the *Suppliers* themselves, by a third party such as the Auditors, or by the blockchain systems like Fabric. Maintaining a fault-tolerant ledger is out of the scope of this paper, but these technologies can be used within it (Xu et al., 2021; Belchior et al., 2022; Ge et al., 2022).

There are *Suppliers* that determine transactions between the nodes and put them by appending transactions into the distributed ledger. The ledger ensures all *Suppliers* and any *Auditor* will scan new transactions. Each entering node and *Auditor* maintains a non-interactive zero-knowledge proof (NIZK) cache sequence, which to values $q = \prod_{i=1}^{n} Sequence\,Q = \beta^{sk\bullet\gamma}$ used to make creating transactions and

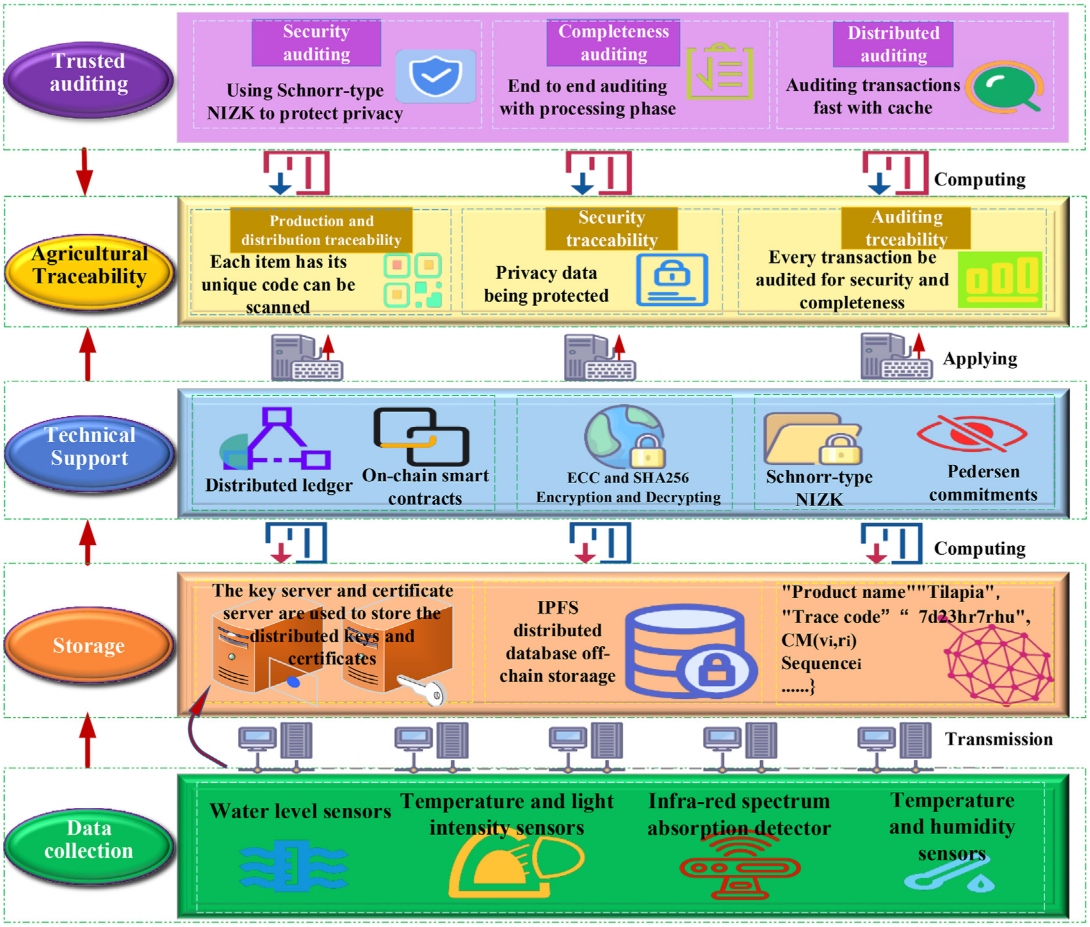

Trusted auditing
Security auditing
Using Schnorr-type NIZK to protect privacy
Completeness auditing
End to end auditing with processing phase
Distributed auditing
Auditing transactions fast with cache
Computing
Agricultural Traceability
Production and distribution traceability
Each item has its unique code can be scanned
Security traceability
Privacy data being protected
Auditing trceability
Every transaction be audited for security and completeness
Applying
Technical Support
Distributed ledger
On-chain smart contracts
ECC and SHA256 Encryption and Decrypting
Schnorr-type NIZK
Pedersen commitments
Computing
Storage
The key server and certificate server are used to store the distributed keys and certificates
IPFS distributed database off-chain storaage
"Product name""Tilapia", "Trace code" " 7d23hr7rhu", CM(vi,ri) Sequencei ......}
Transmission
Data collection
Water level sensors
Temperature and light intensity sensors
Infra-red spectrum absorption detector
Temperature and humidity sensors

Figure 1. Trusted auditing chain overview.

responding to audits faster. Each *Supplier* has its transaction of plaintext content database. Besides, the rest of this section depicts the TA chain transaction form, how the *Suppliers* upload the transactions, and how the *Suppliers* can answer a general audit from the *Auditor*.

3.1.1. Smart contract set up in traceability transaction

In a traditional Fabric traceability blockchain platform, the *Supplier* nodes create a proposal with a user ID, timestamp, chain-code function, and client sign, while selecting an endorsement policy and sending it to the endorsers, which verifies the user's signature and ensuring that the proposal was sent by an authenticated user. The transaction request simulated by the endorsers is executed according to the chaincode selected in the proposal, while the signature of the endorsers is appended to the generated execution result (endorsement process). Thus, the result of the simulation based on the current state of the world is sent together with the endorsement signature to the user, who can verify its consistency. When the verification is successful, then the committers vote on the results submitted by the endorsers (executed using the chaincode), except that the endorsers do not participate in the consensus vote, so they do not know the various results. Eventually, the committers store the write portion of the read-write set off all verified transactions in the state of the world and also update the state database (Ledger A) using the write set. Once the traceability transactions upload into Ledger A, this transaction is committed by every node, immutability maintenance that no one can immunity anything in Ledger A. Ledger A features consensus fairness and by dynamic coordination, each *Supplier* node has a similar probability to be a *Supplier* node for recording signature operations.

It is so difficult to transfer the natural languages to encoding procedures that this work has to consider some issues. Therefore, to ensure the *Auditor* convenience with security auditing, the designed table transaction form with the relevant smart contracts can verify the legal logic during auditing (Figure 2). Considering the legality of the smart contract, it features three characters as follows: (*i*) Useful procedure code. It means the code must be sensitive to certain functions. (*ii*) Present real-world transfer subjective digitally. To audit conventionally, related smart contracts are more willing to present digitally. (*iii*) Designed agreement within the legal context.

3.1.2. Construction set up in traceability transaction

The Ledger A in the TA chain is a cylindrical rolling table where transactions represent *rows*, and supplies correspond to *clumns*. every transaction features context for each *Supplier*. Figure 3 shows each *Supplier* owns its phase and makes the cache to pre-generate the auditing results, which come from the visualized forms below, the private data is Hashed when it is on-chain, and each *Supplier* maintains a public state and its secret key. Simulating the write-set consisting of the transaction for transactions the *Supplier* originated, the private data is Hashed when it is on-chain and hidden from the *Auditor*. This table scheme has a nice display that an outside adversary can look at a *Supplier* entire column and know that this represents the entirety of the *Supplier* information except for privacy. Figure 2 shows an automotive process from the smart contract. For example, it is crucial that the real-time auditing need online, when the new data is appended into the A, the timestamp's settings need to be upgraded, then broadcast to every *Supplier* to verify the transaction (Latency will be taken into account). A transaction needs to meet three requirements below:

- A transaction cannot be tampered with when it was already uploaded, but a transaction with new data can be uploaded as a new transaction.
- A transaction cannot destroy other transactions when auditing.
- Data that has already been uploaded cannot be re-uploaded as a new transaction.

To protect the *Supplier* privacy do not be broadcasted in detail, such as the related private transaction data. Instead, the *Supplier* nodes post the hash of privacy data in the table column and stored the Pedersen commitments (Aranha et al., 2022) value to the append-only Ledger A with its timestamp, especially, Ledger A uses this commitment value to generate cache sequences value. Let G be a cyclic group with ε = |G| elements and let α be a generator and β is a point of Elliptic curves. Then an integer $\nu \in \{0, 1, \ldots, \varepsilon - 1\}$ is formed as a vector, after that, pick randomness γ, and return the cache sequences value $cm = CM(\nu, \gamma) = \alpha^{\nu}\beta^{\gamma}$ mod p, where p is the order of the group G. Some adversaries can be used to compute $\log_{\beta}\alpha$ and thus break the discrete logarithm problem in G. In the TA chain, we choose G to be the group of points on the elliptic curve ECC.

3.2. Non-interactive zero-knowledge proof

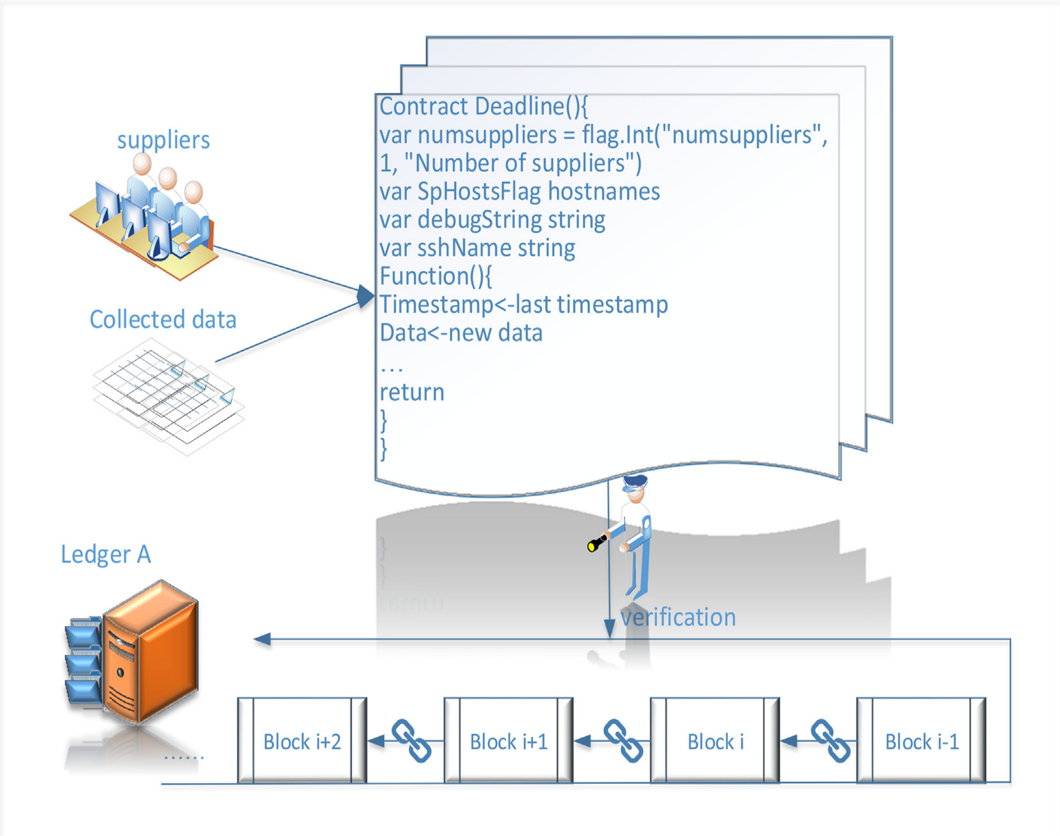
To make a privacy-preserving trusted auditing chain about transaction details, the designed distributed ledger relies on Schnorr-type non-interactive zero-knowledge proofs (Nakamura et al., 2022; Yang and Wang, 2022). Schnorr signatures take up little space and support multiple *Supplier* signature aggregation (multiple participants can work together to produce an aggregated signature based on an aggregated public key), it is beneficial for multiple *Suppliers* making full use of traceability data (Chen and Zhao, 2022; Ciulei et al., 2022). For example, the producer node can prove the private key is owned and show the *Auditor* the signature information which he provided is sent by itself, using Schnorr is the best way to settle both problems synchronously. The Schnorr mechanism also avoids tampering by second parties and is proof-safe, linear, and efficient, even in the case of aggregation, the Schnorr signatures cannot be tampered with unless all signers recreate them. At the same time, Zero-knowledge proof features that one can convey the results of private information without telling the detail of the truth to others so that the sender can pre-vent the private information from adversaries (Wang et al., 2022b), but this method has to be decided by the designer who will choose the proof way. Therefore, the TA chain chooses to use non-interactive Zero-knowledge proof to solve this problem (de Vasconcelos Barros et al., 2022). The great distinguished figure is that the receiver only needs to tell the sender what information they would like to verify, non-interactive Zero-knowledge proof will choose the best approach to get the results and translate them to the sender. For example, the *Auditor* does not have to choose the scope and content of information that they will review, using non-interactive Zero-knowledge proof will compute the bi-nary string α, and the proof will persuade the *Auditor* and does not disclose anything else ν. Moreover, Verifying α does not require any interaction between the *Suppliers* and the *Auditor*, and the *Suppliers* can append α to Ledger A, where it can be verified by any other *Supplier* of the system.

3.2.1. Signature

The Schnorr signature and non-interactive zero-knowledge proof process of the algorithm are described as follows: the first step, is to produce private keys and public keys: Private keys *sk*. Public keys $pk = G^{skn}$ (*G* is a point on the elliptic curve). Secondly, all users produce a randomness number ν respectively which is not spread with others: $\nu_i = G\nu i$, then send the ν_i to the *Auditor*, after the *Auditor collects* all the randomness numbers ($V = \prod_{i=1}^{n} \nu_i$), synchronously, calculating the encoding information $C = H(V \parallel S)$, in which *C* is encrypted transaction information, *H* is a Hash algorithm (SHA256), and *S* is signed information by the sender. The *Auditor* transfers *C* to each user, and comes back *r* to them, so the $r_i = \nu_i - C \cdot ski$. Finally, the *Auditor* collects every r_i: $R = \sum_{i=1}^{n} r_i$, getting the result (*C*, *R*).

3.2.2. Verification

When it comes to authentication, assuming nodes already know the public keys and calculate the $V^{'} = G^{R}\, pk^{C}$, then proving $C = H\,(V^{'} \parallel S)$ is correct, the *C* stands for signature is available. The public keys are created by the Schnorr signature (*pk*, *sk*) and be published to all *Suppliers*. Using NIZK, *Suppliers* will generate a binary string Ω, simultaneously, the



```
Contract Deadline(){
var numsuppliers = flag.Int("numsuppliers",
1, "Number of suppliers")
var SpHostsFlag hostnames
var debugString string
var sshName string
Function(){
Timestamp<-last timestamp
Data<-new data
…
return
}
}
```

Figure 2. Controlling the collected data on-chain with the smart contract and making the legal transactions into Ledger A.

proof will persuade the *Auditor*, yet not disclose anything else about *Suppliers* private information. Verifying Ω does not require any interaction between the *Suppliers* and the *Auditors*, also the *Suppliers* can append Ω to the ledger with a new timestamp, where it can be identified by any party of the TA chain. Consequently, the private key leakage problem of interactive zero-knowledge proofs, which prevents the algorithm from being used in a public environment will be solved.

3.3. Auditing protocol in traceability ledger

The *Auditor* has a copy of Ledger A and the *Supplies* to compute their private data, to get a view of the traceability system represented by the ledger. For example, the *Auditor* checks the *Supplier* node by asking a query to this *Supplier* node, such as the keywords "certification" or "operational personal details". The *Supplier* node responds to the *Auditor* with a processed answer and proves it is true via non-zero knowledge proof, and the *Auditor* gets the answer to calculate the row of transactions which will be consistent with the Ledger A and return it into the ledger restoring complete and consistent.

The deep insight into key issues is that in Ledger A's table construction, the Auditor has access to check the Supplier node accurately because each row features classified information, including public data and private data. The public data can be published to every node and consumer, but the private data should be protected by the other nodes and consumers, the non-zero knowledge proof provides the *Auditor* with a maneuver that has no risk of liability for privacy data breaches. There is no way to hide the privacy information from the *Auditors* in the ledger without actually transferring traceability data and granting control to other Suppliers. On the contrary, the traditional auditing strategy would depict all plaintext to the Auditor without any protection, and would not be conducive to maintaining the security of the traceability system at that such a move when it is not certain that the Auditor is malicious.

As described above, a *Supplier* can propose a transaction online on its own without any previous proclaiming only need a signature in transferring transactions, and it is generally common in blockchain systems. However, Since an *Auditor* cannot determine the transactions proposed by whom, the TA chain must ensure that a participant cannot hide transaction information and even private data from the *Auditor* during the auditing phase, in-including the involved other *Supplier* data, for example, the seller will submit every previous generated blocks' data during auditing phase. Therefore, it is important to privacy-preserving. The TA chain does this by requiring the *Supplier* to include a publicly verifiable Sequence in every entry. Sequence Q is defined as formula1. Supplier uses this Sequence Q to open up the product of its significance for the Auditor, without needing to know r_Q.

In deep insight with each Supplier that they do need to reveal r_i to verify the ν (private data) is true. An assumption that a Supplier calculates $\mathrm{M} = \alpha^{\nu}\beta^{\gamma}$, then the Supplier computes $M' = \mathrm{M}/\alpha^{\nu} = \beta^{\gamma}$, be careful q equals 2. Especially, the Auditor will also calculate the M' and q from the ledger with hashed v. Notice that only when the Auditor figures out the result on 3, then ν can be proved.

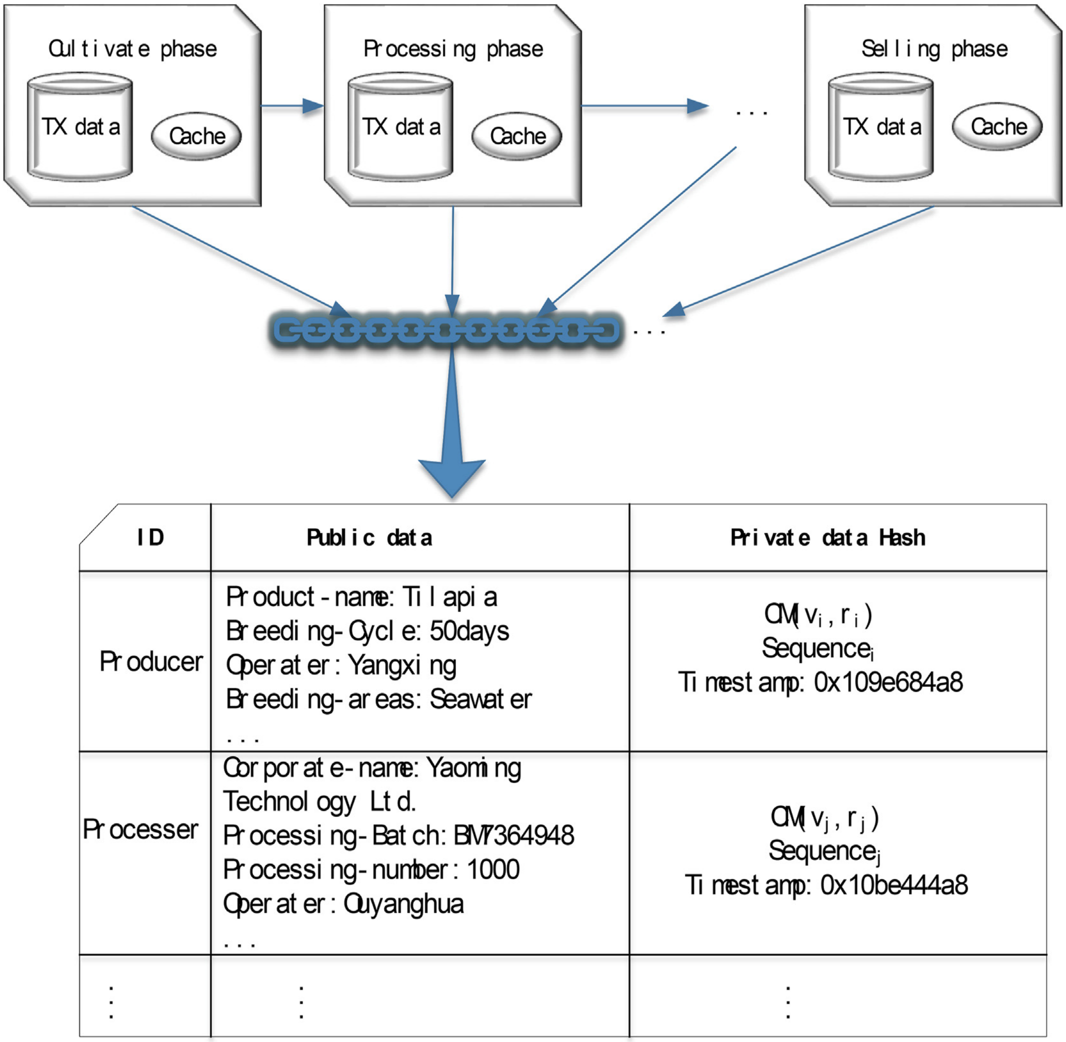
$$\text{Sequence Q} := (pk_i)^{r_i} \tag{1}$$

$$\mathrm{q} = \prod_{i=1}^{n} \text{Sequence Q} = \beta^{sk \bullet \gamma} \tag{2}$$

$$\log_{M'} q = \log_{\beta} pk \tag{3}$$

3.4. Assumption

Users in the TA chain be suggested to store their transactions in a decentralized way. For the sack of the *Auditor* to audit transactions as effectively as they can, to adapt the Ledger A assuming the presence of



ID	Public data	Private data Hash
Producer	Product-name: Tilapia Breeding-Cycle: 50days Operater: Yangxing Breeding-areas: Seawater ...	$CM(v_i, r_i)$ $Sequence_i$ Timestamp: 0x109e684a8
Processer	Corporate-name: Yaoming Technology Ltd. Processing-Batch: BM7364948 Processing-number: 1000 Operater: Ouyanghua ...	$CM(v_j, r_j)$ $Sequence_j$ Timestamp: 0x10be444a8
⋮	⋮	⋮

Figure 3. Data processing in tables for traceability processes.

the experiment is rational. Assume the *Auditor* can be an arbitrary node. Additionally, we make the following assumptions.

3.4.1. Correlation suppliers nodes

This study considers that the majority of nodes are rational and act in a food safety rational manner, such as Block-creators (*Suppliers*) and *Auditors*. In both Bitcoin and Ethereum, peers can automatically identify transactions for the service provided to the decentralized ledger, and the peers can be verified with each other. Therefore, in our permission scenario, making the ideas and repurposing the methodology to set the decentralized network construction is a great way.

3.4.2. Data resources

Basically, the data conveyed into the blockchain network suppose to contain all information about its object, which comes from the local sensors such as scanning imager, Infra-red spectrum absorption detector, humidity transducer, and diverse sensitive resistors, also, the transactions created serially the integrity, and authenticity of data can be ensured, taking the workload off digitizing data, so avoiding mistakes by artificial manufacture.

3.4.3. Storage

Each transaction in the TA chain block features a contractual agreement with the authority that must be met. On-chain transactions are taken charge of each node of the blockchain, the Hash of transactions stored in the block of Ledger A, not all pieces of information will be stored in it, there are additional details involved with some certain missions will be stored off-chain. For example, the pictures and video of produce processing and the off-chain pieces of information of each transaction are stored in a provably correct database, such as IPFS. The physical database storage locations are assumed to be secure.

3.5. Security goals

The goals of the TA chain are to maintain completeness and security auditing with current sensitive information and provide *Auditor* provably auditing approach. TA chain inherits from the Schnorr algorithm the same signature technology to protect transaction data security. Additionally, the TA chain allows *Suppliers* to process their smart contracts and hides the details from the other nodes, every participant can trust a transaction between the *Suppliers*. For example, if the producer node transfers the costs, warehouse stock, and pricing trends, both the producer node and the information are non-plaintext. The public transaction information such as the trade names, growth cycles, food conformity certificates, and individual delivery points are plaintext.

An *Auditor* will query *Suppliers* such as producers about its content respectively, for example, “check the producer certificates”. The producer is supposed to make a response that will convince the *Auditor* the answer is the same as the Ledger A and not encounter fraud. Additionally,

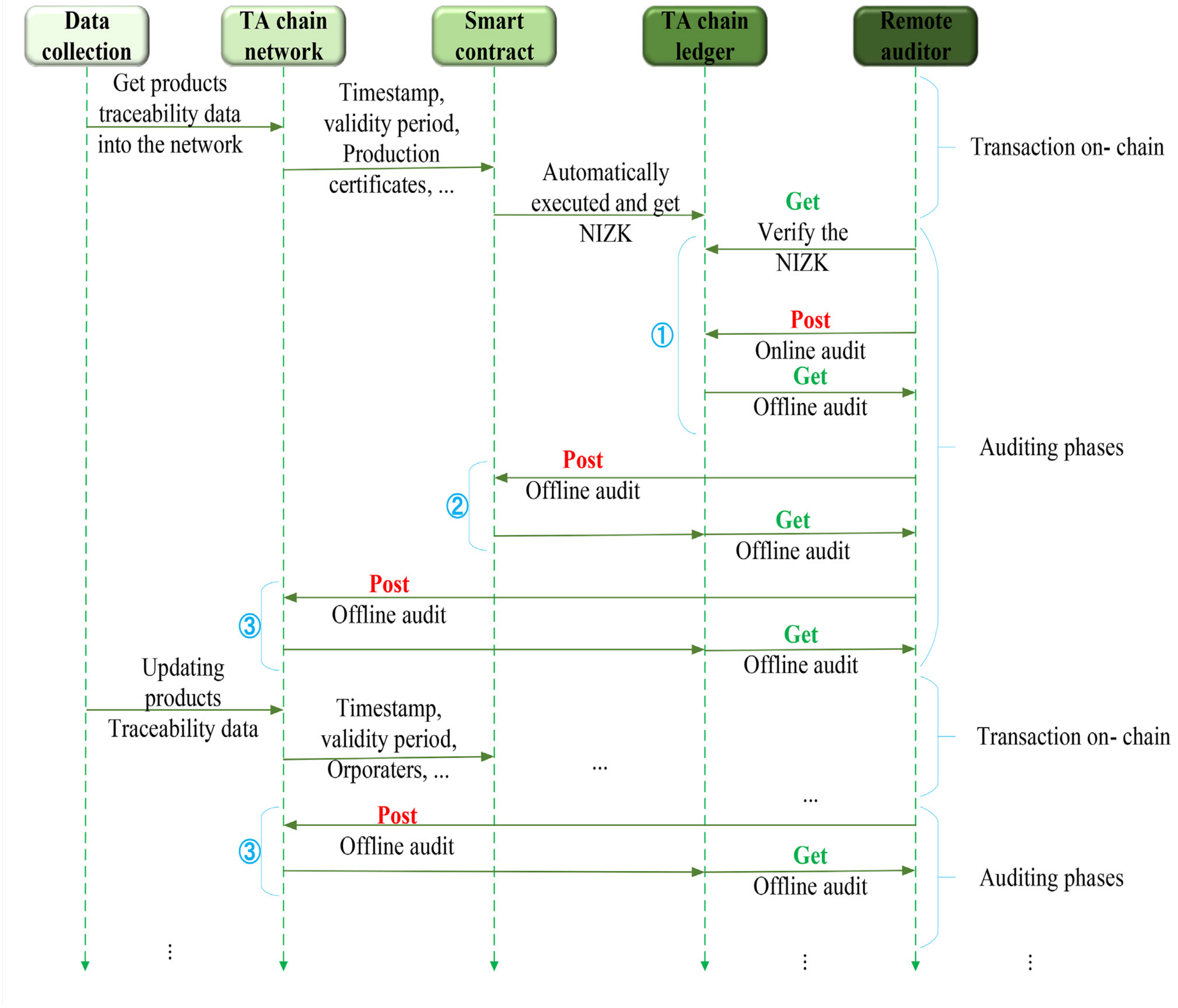


Figure 4. NIZK Integrity in Trusted auditing chain.

to verify the legal logic, the TA chain also audits transactions with smart contracts. On the contrary, if the answer of producers does not match Ledger A, then the *Auditor* will discover this cheat with them and make an illegal record. Correspondingly, smart contracts will give a response to illegal acts, such as rollback.

4. Implementation and evaluation

To evaluate the preference of the TA chain, this study de-signed a simple traceability prototype for implementation in 1.16.5*darwin/amd*64, using a computer with 2 GHz quad-core Intel Core i5 and 16G of 3733 MHz RAM. The infrastructure of the permissioned blockchain is based on the Hyperledger Fabric V2.2. Original transactions will be processed by the SHA256 cryptographic hash function, this act can maintain the structure reunion and avoid being tampered with, the certificates and the keys are stored in their servers. A general approach is used in this prototype, which includes the values and methods to compute with the elliptic curve ECC (using very little bandwidth and storage resources, having a short key length, and allowing all *Suppliers* to use the same operation for domain operations).

From previous work, the study proposes a new integrated approach to stabilize the private preserving blockchain which can provide security auditing (Figure 4), the 2 and 3 sequences at the Auditing phases can equal progress synchronously for the Ledger A and the *Auditor*.

4.1. Implementation of trusted auditing chain

To test the full process against a running prototype of the TA chain, and doing so allows for a better understanding and maintenance of the real participating users, thus, promoting product and servicing designs that are friendly to both users and regulators. Furthermore, the scientific approach to design deserves effective improvement experiments, as it involves algorithms to optimize the process of existing solutions, as well as developing new solutions for new problems that may arise.

To ensure that the data is valid in real-time, we process the data uploaded from the sensors (Figure 1), uploading the data from the bottom data collection to a remote server, which uses the upper layer privacy protection technology to get the processed data into our permissioned blockchain, the processing is completed and deposited into the form, the information that needs to be traced is visible in plaintext (Figure 2). Of course, a smart contract needs to be triggered before this to endorse the transaction uploaded (Figure 4), but a column involving private data must be protected and requires direct computing on private data, just to be used when the regulator is auditing, and after completing the audit of the uppermost layer, the experimental design of the trusted audit chain is complete. The auditing time of the trusted audit chain can be arbitrary, to prevent malicious attacks, optimizing the algorithm is necessary so data can be audited offline and can also be audited securely when it is online, mainly with the help of zero-knowledge proofs.

The communication of this network requires a large number of time. Before the auditing, if a producer attempts to implement a computational task as possible it can, it must control the other side of the network, then the trusted auditing can be sure that when the *Auditor* works with the TA chain will not change the environment of variances (Table 2). The considered network latency is coming from a large number of transaction submission times and auditing computing time. Therefore, there are two conditions which are online auditing and offline auditing.

4.1.1. Online

The data in the ledger is original information from the live-site collection. Therefore, smart contracts take a critical role in this process, each transaction always fulfills the conditions of endorsers during the consensus. The *Auditor* can check the online entry information as soon as they can. For example, each *Supplier* stores rolling results of the computing by row and by private level to quickly produce privacy values and answer queries from the *Auditor*. in a *Supplier* column, caching the results of the computing improves auditing and proof calculation speed significantly. Using these cache sequences, a *Supplier* can quickly respond to the *Auditor* asking about a subset of rows in the ledger. The designed TA chain Curve is a global cache sequence for the elliptic curve and two generator points are used in the various proof, the generated points can be used to produce a sufficiently secure key pair which is sure the Auditor verifies the results of NIZK. Hence, online auditing is important for the TA chain.

4.1.2. Offline

Many traceability data are willing to be stored offline, on the one hand, considering the communication speed, cache sequences, and throughputs. On the other hand, food information always needs highly sensitive time so the real-time update and audits are significant, and new food transactions need new timestamps to verify. Hence, unfresh data or additional information can be stored offline. But the *Auditor* must tackle the whole ledger to finish completeness auditing, thus this involved more complex auditing when the *Auditor* has to check every row's entry. Meanwhile, though online auditing is critical for traceability information to publish, much-existing data is stored offline, and most smart contracts can implement online automatically but are not offline running. Therefore, offline auditing is also necessary for food safety traceability.

4.2. Evaluation of trusted auditing chain

To show the clear consequence, every point runs the query 10 times in these figures. Table 2 shows the other prerequisite of the distributed ledger start-up.

Table 2. The TA chain network performance evaluation configuration.

Experiment Parameters	Single supply	Multiple suppliers
Number of supply	1	1–10
Number of transaction	16–1024	64/512
Number of *Auditor*	1	
Permissioned ledger	1	
Encrypted channel	1	
Number of cycles audited	10	
Print progress of results/ms	60	
Remote basic-port	8000	

4.2.1. Audit time

Average auditing time depicts the distinction between with cache and without cache. The first group (Figure 6) fulfills index growth transactions, the *Auditor* would need better control of variances and an idea status, when the number of transactions is growing, the offline will spend much time asking for the answers. All data in the TA distributed ledger is under NIZK protection. In consideration of multiple *Suppliers* audited, con-auditing needs high CPU computing capacity, and also meets the ledger storage features. To test the scalability of the TA chain, we grew the number of individual supplier transactions from 16 to 1024. This indicates that with the increasing number of *Suppliers* both transaction creation and verification times per supply node increase linearly, but nodes increasing helps parallelization.

When the *Auditor* cannot audit through the cache, it means that it is offline at the time, so it must calculate the transactions across the chain to audit the current results. This can also be applied to more complex auditing jobs. This graph shows the comparison of the time required with and without the use of cache auditing (Figure 6), and as expected the time cost increases linearly and the results are reasonable even without the use of cache calculations. the TA chain currently only maintains a cache of transactions from each supplier, but it could maintain many more.

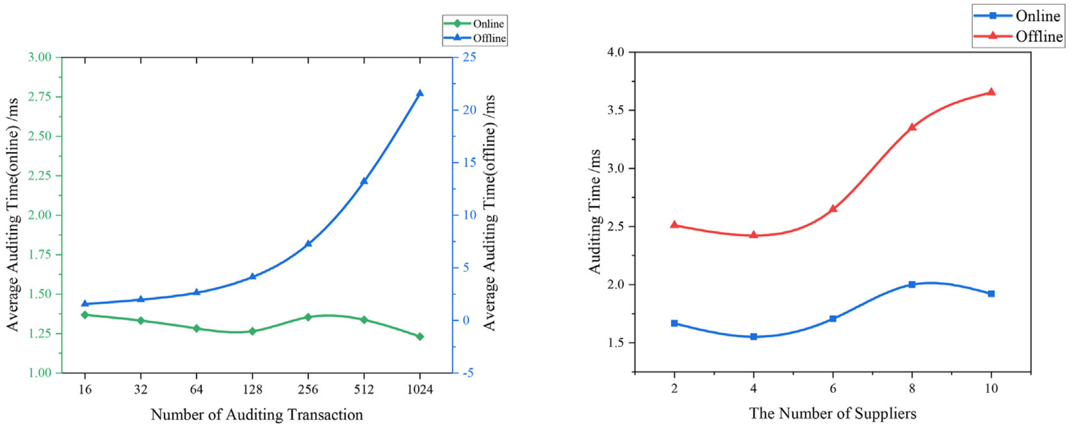
4.2.2. Audit throughput and the cost

In this study, throughput is used to represent the ability of the TA chain to handle concurrent operations. In a Hyperledger blockchain-based system, the throughput of TA chain auditing relies heavily on the throughput of transactions and the number of providers processing transactions simultaneously. We measure transaction throughput by examining the relationship between the concurrency values of transactions and the speed at which these transactions are processed. For a more comprehensive analysis, the time of transaction generation and validation is kept in dispute and the probability of each Supplier having a malicious act is assumed to be equal.

For a fixed-size ledger, this audit function requires several *Suppliers* to test the robustness of the TA chain. The graph on the right shows the cost of the audit calculated on a ledger of 64 as we change the number of *Suppliers*, both with and without caching (Figure 5). The *Auditor* audits the *Supplier* at the same time. The audit cost for this function increases slightly with the number of *Suppliers* because more of them increase the variability of the parallel audit and the *Auditor* must wait for the last transaction to respond before calculating a final answer.

As Figure 5 depicts the system throughput under auditing requests, and the system throughput stabilizes at about 60 when the auditing requests increase by more than 1000. When checking the latency experience, with the transaction growing, each turn takes more than 16 ms less than 18 ms time to audit transactions, which means the latency does not change much with transaction growth, because of the network's robustness. Meanwhile, with the increase in the *Suppliers*, transactions can be verified in parallel, so the time clamp linearly.

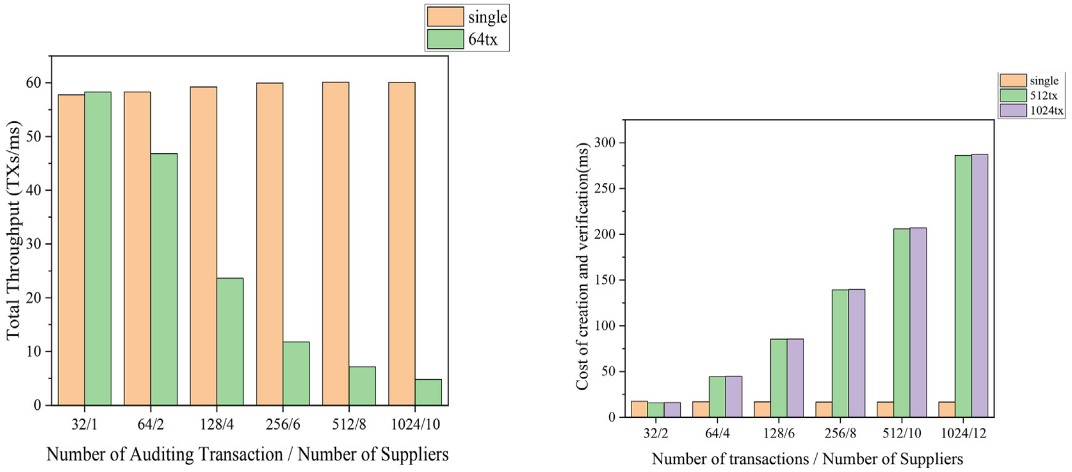
The first two sets of graphs on the left, though there is also latency (time for transaction generation and validation) when only one node is working in the ledger, so the throughput is less than 58.5 even TX $= 64$ (as an input), and the throughput is less than 60 when a *Supplier* $= 1$ in the multi-node graph, the reason it is not the same is that throughput fluctuates with the quality of the network, although every effort has been made to preserve the same environment for each test. After TX grows to 256, the offline auditing time increases faster, while the online plateaus. Combined with Figure 5(a), TX > 256 is nearly flat afterward. Thus, on balance, a transaction throughput of 256–512 is optimal for both online and offline auditing when *Supplier* $= 1$. Meanwhile, comparing with the first two sets of graphs on the right, we also dis-cuss the pattern of node increase in this permissioned blockchain and find that the rate of online and offline auditing is optimal when the *Supplier* nodes are at 6.



(a)

(b)

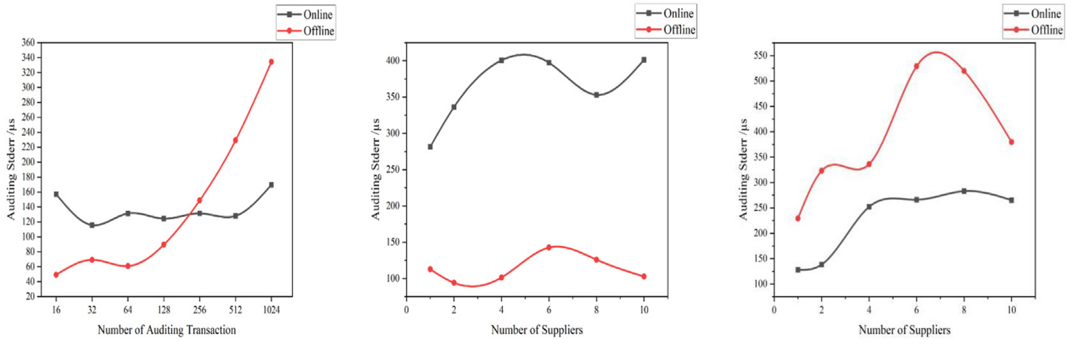
Figure 5. This work sets the growing auditing transaction number as exponential multiplier growth, with the rapid transaction increase of more than 1000, the online auditing is stable within the changing scape of 0.01 ms, on the contrary, the offline auditing needs 21 ms at least (a). Because online auditing uses the table cache to pre-generate auditing results if the Auditor asks, pre-generate auditing results can be a conserved response time. Setting the TXs are 64, the online auditing time is still much more stable than offline auditing time, even though the multiple Suppliers (b).



(a)

(b)

Figure 6. The total throughput decreases as the number of TXs and Suppliers increases (a), because throughput restricts the auditing time of multi-suppliers (latency can also tell), then compare with (a) the auditing time increases apparently, except for online auditing. Creation and validation costs have proven to exist, but are acceptable (b), 'single' means just only a Supplier, but TXs are increasing from 32 to 1024. The other two both represent the multi-Suppliers case, green means TXs = 512~, and purple means TXs = 1024.



(a)

(b)

(c)

Figure 7. Standard deviation error (Stderr) means the unstable scenario during the auditing phase, (a) is a single Supplier, (b) and (c) are multiple Suppliers which be audited by 64 TXs and 512 TXs.

4.2.3. Accuracy evaluation

As communicating and validating transactions dominates the cost of transaction creation and validation, the study affirms that a faster online audit implementation will directly improve performance. This study uses standard deviation error (Stderr) to verify the stability of online auditing and offline auditing during the audit phase (Figure 7). As the number of transactions increases (Figure 7(a)), the Stderr for the offline scenario rises sharply, although it is lower at the beginning than in the online scenario. In contrast, considering online auditing, where the variation interval lies between 170 ms and 110 ms, the Stderr is relatively stable, implying that individual vendors are not constrained by transactions and that their cached audits contain a good computational environment.

In general, because of the instability of offline auditing, there is an inflection point in the graph on the right, which we analyze as a loss of communication costs (transmission and validation) due to simultaneous remote auditing, where the error suddenly increases when the number of suppliers audited simultaneously increases ($6 < Suppliers < 8$), and slowly stabilizes but remains larger than before, as the number of suppliers audited simultaneously continues to increase. The simultaneous throughput then decreases. This result is in line with the expectations of our research scenario, and therefore the number of controlled audit transactions is controlled at around 512 to achieve optimal audit efficiency.

Thus, while a single *Supplier* can use multiple cores to generate proofs of individual transactions in parallel, multiple *Suppliers* cannot generate different transactions in parallel. The major cost relates to creation and validation. Each *Supplier* must verify each created transaction, so the more *Suppliers* there are, the larger each transaction becomes, and therefore the more work each *Supplier* needs to do (Figure 5). Because of latency, proving and validating dominates transaction creation and verification, but this cost is also highly parallelizable. They measure in Figure 6, the time it takes from the creation to the audit of a *Supplier* and all the participants in the TA chain to fully process a transaction. One of the *Suppliers* creates a transaction and sends it to the ledger, it then broadcasts the transaction to all *Suppliers* and an online *Auditor*, both online and offline. Both the *Auditor* and each *Supplier* have to verify the transaction as the number of *Suppliers* increases, the work increases quadratically, but the *Suppliers* can verify the transaction in parallel, so the time to process the transaction increases only linearly.

4.2.4. Performance comparison

Here, we assume that run 1000 transactions per second with a latency of less than 1 s using PBFT consensus (Belchior et al., 2022). This is a reasonable assumption from the perspective of the widely used Etherand Hyperledger. Then, we compare the TA chain with other blockchain systems, considering the other parameter settings designed in the two audit approaches compared to the TA chain (Table 3).

Because of the importance of the security of blockchain, the permissioned blockchain is the best choice for the TA chain system. Because permissioned blockchain is suitable for agribusiness supply chain needs that are more private and confidential, in-stead public blockchain is more transparent in traceability and is suitable for fair trade in agribusiness. Finally, there is a high-level comparison between other blockchain systems and the TA chain from some vital aspects. Something needs to be noted that, as the property of centralized regulation, Bitcoin provides decentralized decision-making by community/miners (Nakamoto, 2008), Ethereum is decided by the core developer group (Dannen, 2017), Hyperledger is an open-governance model (Ge et al., 2022), and the TA Chain is an auditing regulation model. In addition, considering the property of scripting, Bitcoin only provides stack-based scripting (i.e., C++), Ethereum proposes a Turing-complete virtual machine, high-level language support (i.e., Solidity), and both Hyperledger and TA chain own Turing-complete scripting of chaincode (i.e., high-level Go-language).

Table 3. Comparison between other blockchain systems and the TA chain.

Performance	Bitcoin	Ethereum	Hyperledger	TA chain
Type of blockchain	Public	Public	Permissioned	Permissioned
Transaction fee	Yes	Yes	No	No
Consensus protocol	PoW	PoW	PBFT	PBFT
Anonymity	Pseudo	Pseudo	Pseudo	Pseudo
Transparent	High	High	Medium	Medium
Privacy of transaction data	No	No	Yes	Yes
Scripting	C++	Solidity	Golang	Golang

4.3. Threat analysis

Besides the above results being given to us, another unfavorable condition is considered.

4.3.1. Threat attack

The results cannot be sure the *Suppliers* side hardware will be trusted by all nodes, if the *Suppliers* do not trust this hardware, interest conflict is inevitable. In addition, introducing third-party scientific auditing exist a potential threat of attack because the *Auditor* identity cannot be identified to every *Supplier* node, a permissioned blockchain would remit the threat but not eliminate it. Moreover, if a *Supplier* node can operate a transaction collision when it acts as an *Auditor* on the transaction, the *Supplier* can deny its operation. As far as it is concerned, a distributed digital signature would do its work instead of authorities confirming.

4.3.2. Rollback

However, there is a negative condition in the auditing protocols. If the transaction does not match the smart contracts, then be rollback to the initial statute, which would unworthy of auditing. But can slow down the transfer speed, occupancy cache, and the resource of throughputs. To promote the auditing rate, the ineffective transaction entries must be solved. Maybe the next study will start to solve this problem to arise transactions effectively. Chaincodes are used for to consent transaction process, which maybe is not available but the system has no idea about that. We know the transition is down but whether the transaction is available and correct or not is uncertain. Most traditional smart contracts usually incorporate financial transactions of digital assets so that they must be highly safe and trustworthy, but for agricultural traceability, it needs to consider another problem, such as completeness auditing.

4.3.3. Dceive

The *Suppliers* can also be regarded as an *Auditor* to audit other *Suppliers* data, here is the security problem. It is afraid that a $Supplier_i$ colludes with $Supplier_j$, then exchanges the transmitted information. If a malicious vendor node wants to pretend that it is an *Auditor* to audit other nodes, the encrypted private data and the rigorous mechanism of zero-knowledge proof make it impossible for that malicious node to get the private information of the inspected node, and it can only see the traceable public information and get the proof information that the node is legitimate.

TA chain focuses on providing provably secure auditing of food traceability transaction data, but if the distributed ledger is corrupted, the TA chain can only recover it through an offline database. In this case, the ledgers of record for multiple parties must be brought together to recreate the historical transactions. Future versions of the TA chain may provide for corrected transactions or participant-agreed rollbacks.

5. Conclusion

The Trusted Auditing chain (TA chain) is related to work in auditing or computing private data of traceability and privacy-preserving blockchains. TA chain aims to effectively, correct auditing by creating a new decentralized ledger auditing table model and applying a new scheme using zero-knowledge proofs. We give an overview of the fundamental properties that a blockchain-based trusted auditing system must satisfy.

Based on such properties, making a comparison of the auditing features between the TA chain and the existing systems. For example, balancing the data calculation only on private data or on the whole blockchain ledger, also fairing the auditing scope whether includes decentralized and fair auditing or the extent to security and completeness auditing. Finally, we find that most existing works would like to compute on private data but not set security mechanisms in the blockchain ledger. Therefore, they may use blockchain to finish the distributed audit, but cannot be sure of its security and completeness of auditing. Therefore, a new idea which is integrity private preserving into the trusted auditing chain can not only guarantee the sensitive information but also convince the transactions are true and stored in this distributed ledger.

In this study, the proposed security audit shows oversight of the privacy and integrity of agri-food traceability systems. In particular, the work uses non-interactive zero-knowledge proofs to validate transactions even without acknowledging some im-important private data and uses Pedersen's promise to pre-generate cached values to reduce audit latency and optimize the stability of audit processing. By comparing the feasibility and efficiency of offline and online auditing, we find that remote auditing is stable with good communication conditions and both can be performed, but in the case of high communication and latency, auditing with caching can be chosen, which can improve the efficiency of auditing. In addition, our research can also help to integrate other traceability systems that require privacy-preserving audits, such as fruit and vegetable traceability audits or cold chain audits. The theoretical analysis and experimental results of this experiment show that the proposed scheme has good performance. The experiments in our paper are currently considering data from the agricultural traceability sector, but it is something that can be actively extended. As mentioned above, testing for security cannot guarantee collusion between Suppliers and Auditors, and the security of smart contracts is also an issue that needs to be considered, which is also something we will be looking at in the future.

Declarations

Author contribution statement

Lei moyixi: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Liu shuangyin; Yang xinting: Contributed reagents, materials, analysis tools or data.

Luo na; Sun chuanheng: Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Funding statement

Shuangyin Liu was supported by Guangzhou Science and Technology plan Project (202103000033), Natural Science Foundation of china (61871475).

Professor chuanheng sun was supported by National Natural Science Foundation of China (31871525).

Data availability statement

The authors do not have permission to share data.

Declaration of interest's statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

Khanfar, Ahmad AA., Iranmanesh, Mohammad, Ghobakhloo, Morteza, Senali, Madugoda Gunaratnege, Fathi, Masood, 2021. Applications of blockchain technology in sustainable manufacturing and supply chain management: a systematic review. Sustainability 13 (14), 7870.

Aranha, Diego F., Bennedsen, Emil Madsen, Campanelli, Matteo, Ganesh, Chaya, Orlandi, Claudio, Takahashi, Akira, 2022. Eclipse: enhanced compiling method for pedersen-committed zksnark engines. In: IACR International Conference on Public-Key Cryptography. Springer, pp. 584–614.

Belchior, Rafael, Vasconcelos, Andre ́, Correia, Miguel, Hardjono, Thomas, 2022. Hermes: fault-tolerant middleware for blockchain interoperability. Future Gener. Comput. Syst. 129, 236–251.

Bonde Thylstrup, Nanna, Archer, Matthew, Ravn, Louis, 2022. Traceability. Internet Policy Rev. 11 (1), 1–12.

Chen, Yanbo, Zhao, Yunlei, 2022. Half-aggregation of Schnorr Signatures with Tight Reductions. Cryptol. ePrint Arch.

Chenli, Changhao, Tang, Wenyi, Frank, Gomulka, Jung, Taeho, 2022. Provnet: networked bi-directional blockchain for data sharing with verifiable provenance. J. Parallel Distr. Comput. 166, 32–44.

Dannen, Chris, 2017. Introducing Ethereum and Solidity, vol. 1. Springer.

de Vasconcelos Barros, Mauricio, Schardong, Frederico, Custo ́dio, Ricardo Felipe, 2022. Leveraging self-sovereign identity, blockchain, and zero-knowledge proof to build a privacy-preserving vaccination pass. arXiv. Preprint arXiv:2202.09207.

Dodd, Meaghan, Intuitive Food Solutions, 2022. Discussion Paper on Seafood Traceability. Intuitive Food Solutions, pp. 1–33.

Francati, Danilo, Ateniese, Giuseppe, Faye, Abdoulaye, Maria Milazzo, Andrea, Perillo, Angelo Massimo, Schiatti, Luca, Giordano, Giuseppe, 2021. Audita: a blockchain-based auditing framework for off-chain storage. In: Proceedings of the Ninth International Workshop on Security in Blockchain and Cloud Com-Puting, pp. 5–10.

Ge, Zerui, Loghin, Dumitrel, Ooi, Beng Chin, Ruan, Pingcheng, Wang, Tianwen, 2022. Hybrid blockchain database systems: design and performance. VLDB Endowment 15 (5), 1092–1104.

Ateniese, Giuseppe, Burns, Randal, Curtmola, Reza, Herring, Joseph, Kiss- ner, Lea, Peterson, Zachary, Song, Dawn, 2007. Provable data possession at untrusted stores. In: Proceedings of the 14th ACM Conference on Computer and Commu- Nications Security, pp. 598–609.

Brebu, Andrei-Alexandru, Iacov, Mihai, Simion, Emil, 2022. Storage Security in Cloud Computing: Data Auditing Protocols. Cryptology ePrint Archive.

Jin, Hao, Jiang, Hong, Zhou, Ke, 2018. Dynamic and public auditing with fair arbi- tration for cloud data. IEEE Trans. Cloud Comput. 6 (3), 680–693.

Yang, Kang, Wang, Xiao, 2022. Non-interactive zero-knowledge proofs to multiple verifiers. Cryptol. ePrint Arch.

Konkin, Anatoly, Zapechnikov, Sergey, 2021. Privacy methods and zero-knowledge poof for corporate blockchain. Procedia Comput. Sci. 190, 471–478.

Hang, Lei, Ullah, Israr, Kim, Do-Hyeun, 2020. A secure fish farm platform based on blockchain for agriculture data integrity. Comput. Electron. Agric. 170, 105251.

Lei, Moyixi, Xu, Longqin, Liu, Tonglai, Liu, Shuangyin, Sun, Chuanheng, 2022. Integration of privacy protection and blockchain-based food safety traceability: potential and challenges. Foods 11 (15), 2262.

Liu, Qin, Wang, Guojun, Wu, Jie, 2014. Consistency as a service: auditing cloud consistency. IEEE Trans. Network Service Manag. 11 (1), 25–35.

Liu, Feng, Feng, Zhefu, Qi, Jiayin, 2022a. A blockchain-based digital asset platform with multi-party certification. Appl. Sci. 12 (11), 5342.

Liu, Feng, Yang, Cheng-yi, Yang, Jie, Kong, De-li, Zhou, Ai-min, Qi, Jia-yin, Li, Zhi-bin, 2022b. A hybrid with distributed pooling blockchain protocol for image storage. Sci. Rep. 12 (1), 1–10.

Nakamoto, Satoshi, 2008. A peer-to-peer electronic cash system. Decentr. Business Rev. 4, 21260.

Nakamura, Makoto, Miyamae, Takeshi, Morinaga, Masanobu, 2022. A privacy- preserving outsourcing scheme for zero-knowledge proof generation. J. Inf. Process. 30, 151–154.

Perera, P.A.S.N., Abeygunasekera, A.W.J.C., 2022. Blockchain adoption in accounting and auditing: a qualitative inquiry in Sri Lanka. Colombo Business Journal 13 (1).

Rabaninejad, Reyhaneh, Rajabzadeh Asaar, Maryam, Attari, Mahmoud Ahmadian, Reza Aref, Mohammad, 2020. An identitybased online/offline secure cloud storage auditing scheme. Cluster Comput. 23 (2), 1455–1468.

Ruj, Sushmita, Rahman, Mohammad Shahriar, Basu, Anirban, Kiyomoto, Shinsaku, 2018. Blockstore: a secure decentralized storage framework on blockchain. In: 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA). IEEE, pp. 1096–1103.

Ciulei, Andrada-Teodora, Creţu, Marian-Codrin, Simion, Emil, 2022. Preparation for post-quantum era: a survey about blockchain schemes from a post-quantum perspective. Cryptol. ePrint Arch. 1–38.

Wang, Lixing, He, Yulin, Wu, Zhenning, 2022a. Design of a blockchain-enabled traceability system framework for food supply chains. Foods 11 (5).

Wang, Zhipeng, Chaliasos, Stefanos, Qin, Kaihua, Zhou, Liyi, Gao, Lifeng, Berrang, Pascal, Livshits, Ben, Gervais, Arthur, 2022b. On How Zero-Knowledge Proof Blockchain Mixers Improve, and Worsen User Privacy. arXiv. Preprint arXiv:2201.09035.

Wang, Yujue, Wu, Qianhong, Qin, Bo, Tang, Shaohua, Susilo, Willy, 2017. Online/offline provable data possession. IEEE Trans. Inf. Forensics Secur. 12 (5), 1182–1194.

Wang, Chao, Chen, Shizhan, Feng, Zhiyong, Jiang, Yanan, Xue, Xiao, 2019a. Blockchain-based data audit and access control mechanism in service collaboration. In: 2019 IEEE International Conference on Web Services (ICWS). IEEE, pp. 214–218.

Wang, Huaqun, He, Debiao, Jia, Yu, Wang, Zhiwei, 2019b. Incentive and unconditionally anonymous identity-based public provable data possession. IEEE Trans. Serv. Comput. 12 (5), 824–835.

Wang, Jia, Peng, Fang, Tian, Hui, Chen, Wenqi, Lu, Jing, 2019c. Public auditing of log integrity for cloud storage systems via blockchain. In: International Conference on Security and Privacy in New Computing Environments. Springer, pp. 378–387.

Xie, Zhenjun, Kong, Hua, Wang, Bin, 2022. Dual-chain blockchain in agricultural e-commerce information traceability considering the viniar algorithm. Sci. Program. 2022, 1–10.

Xu, Minghui, Liu, Chunchi, Zou, Yifei, Zhao, Feng, Yu, Jiguo, Cheng, Xiuzhen, 2021. wchain: a fast fault-tolerant blockchain protocol for multihop wireless net- works. IEEE Trans. Wireless Commun. 20 (10), 6915–6926.

Xue, Jingting, Xu, Chunxiang, Zhao, Jining, Ma, Jianfeng, 2019. Identity-based public auditing for cloud storage systems against malicious auditors via blockchain. Sci. China Inf. Sci. 62 (3), 1–16.

Yu, Haiyang, Yang, Zhen, Tu, Shanshan, Waqas, Muhammad, Liu, Huan, 2022. Blockchain-based offline auditing for the cloud in vehicular networks. IEEE Trans. Network Serv. Manag. 1–13.

Zhu, Xuecheng, Yuan, Xinyue, Zhang, Ying, Liu, Huilin, Wang, Jing, Sun, Baoguo, 2022. The global concern of food security during the covid-19 pandemic: impacts and perspectives on food security. Food Chem. 370, 130830–130830.