A Blockchain-Based Decentralized, Fair and Authenticated Information Sharing Scheme in Zero Trust Internet-of-Things

Yizhi Liu, Xiaohan Hao, Wei Ren, *Member, IEEE*, Ruoting Xiong, Tianqing Zhu, *Member, IEEE*, Kim-Kwang Raymond Choo, *Senior Member, IEEE*, and Geyong Min, *Member, IEEE*

Abstract—Internet-of-Things (IoT) are increasingly operating in the zero-trust environments where any devices and systems may be compromised and hence untrusted. In addition, data collected by and sent from IoT devices may be shared with and processed by edge computing systems, in order to reduce the reliance on centralized (cloud) servers, leading to further security and privacy issues. To cope with these challenges, this paper proposes an innovative blockchain-enabled information sharing solution in zero-trust context to guarantee anonymity yet entity authentication, data privacy yet data trustworthiness, and participant stimulation yet fairness. This new solution is able to support filtering of fabricated information through smart contracts, effective voting, and consensus mechanisms, which can prevent unauthenticated participants from sharing garbage information. We also prove that the proposed solution is secure in the universal composability framework, and further evaluate its performance over an Ethereum-based blockchain platform to demonstrate its utility.

Index Terms—Blockchain, information sharing, zero-trust, Internet-of-Things, privacy protection, smart contract

1 Introduction

Internet-of-Things (IoT) devices, such as consumer IoT devices (e.g., inexpensive home IoT devices and more computationally capable IoT devices), medical IoT devices, and industrial IoT devices, are increasingly commonplace. For example, it was estimated that the number of IoT devices connected to the Internet will be approximately 75 billion by 2025 [1]. IoT devices are also getting more sophisticated, for example, in terms of enhanced storage capability and functionalities; thus, allowing broader range and type of data to be sensed and collected. Such data can then be used or shared to facilitate various data analytical applications, for instance, to facilitate data-driven decision making. Increasingly, IoT information sharing is decentralized (e.g., peer-to-peer – P2P), and this reinforces the importance of entity authentication with or without the involvement of some trusted third-party (TTP).

Zero-trust is also a relatively recent security requirement, since IoT devices and systems may operate in adversarial environments where other devices and systems cannot be trusted. Hence, a recent research trend is to design authenticated information sharing over decentralized and untrusted (e.g., zero-trust) environments. In addition to security, we stress the importance of ensuring privacy-awareness in shared information, as well as fairness. While there have been attempts to utilize blockchain in the design of systems that provide some combinations of decentralization, privacy-awareness, fairness, behavior verification, identity authentication, sharing tracing, and penalty enforcement for information-sharing in zero-trust settings (e.g., [2], [3]), there is no solution that supports all these properties. This motivates our work reported in this paper.

In this paper, we propose a blockchain-based information-sharing protocol designed to operate in a zero-trust environment. For example, IoT nodes (e.g., onboard sensors on autonomous vehicles) can share information autonomously with other IoT nodes, instead of uploading the

- *Yizhi Liu, Ruoting Xiong, and Tianqing Zhu are with the School of Computer Science, China University of Geosciences, Wuhan 430078, China. E-mail: 123654784@qq.com, 20161003391@cug.edu.cn, tianqing.zhu@ieee.org.*
- *Xiaohan Hao is with the School of Software, University of Technology Sydney, Ultimo, NSW 2007, Australia, and also with the School of Computer Science, China University of Geosciences, Wuhan 430078, China. E-mail: xhhao@cug.edu.cn.*
- *Wei Ren is with the School of Computer Science, China University of Geosciences, Wuhan 430078, China, and with the Yunnan Key Laboratory of Blockchain Application Technology, Kunming 650500, China, and also with the Hubei Key Laboratory of Intelligent Geo-Information Processing, Wuhan 430074, China. E-mail: weirencs@cug.edu.cn.*
- *Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249 USA. E-mail: raymond.choo@fulbrightmail.org.*
- *Geyong Min is with the Department of Computer Science, University of Exeter, EX4 4QF Exeter, U.K. E-mail: G.Min@exeter.ac.uk.*

Manuscript received 28 Oct. 2021; revised 2 Feb. 2022; accepted 6 Mar. 2022. Date of publication 9 Mar. 2022; date of current version 13 Jan. 2023.
This work was supported in part by the Foundation of Yunnan Key Laboratory of Blockchain Application Technology under Grants 202105AG070005 and YNB202103, in part by the National Natural Science Foundation of China under Grant 61972366, in part by the Provincial Key Research and Development Program of Hubei under Grant 2020BAB105, in part by the Foundation of Henan Key Laboratory of Network Cryptography Technology under Grant LNCT2020-A01, and in part by the Foundation of Hubei Key Laboratory of Intelligent Geo-Information Processing under Grant KLIGIP-2021B06. The work of Kim-Kwang Raymond Choo was supported only in part by Cloud Technology Endowed Professorship.
(Corresponding author: Wei Ren.)
Recommended for acceptance by X. Jia.
Digital Object Identifier no. 10.1109/TC.2022.3157996

information to a centralized server. Building on this basic model, we design a blockchain-based authentication protocol to authenticate ad-hoc information sharers, and a smart contract-based mechanism to detect and filter potentially fabricated information. In other words, our approach prevents unauthenticated participants from sharing incorrect information and ensuring the authenticity and fairness of data by deterring reasonable participators, using voting and penalty. The capability to impose and enforce penalties will discourage participants from inserting fabricated information, and a misbehaving user can be delisted from the system.

A summary of our approach is as follows:

- We propose a blockchain-enabled decentralized information-sharing protocol that is designed to operate in the zero-trust IoT environment, where participants can autonomously complete the sharing process independently without relying on a TTP. Additionally, our protocol is proven to be secure in the universal composability framework.
- Our proposed scheme can guarantee fairness by detecting and filtering fabricated information through smart contracts. We also design an effective voting mechanism with in-built consensus and penalty capabilities. Thus, a misbehaving user can be penalized and blacklisted.
- Our proposed scheme can guarantee entity authentication with privacy protection for their identity. Also, our approach supports the distribution of temporary keys, which can be used to encrypt sensitive information (e.g., geolocations).

The remainder of this paper is organized as follows. Section 2 presents the system model and design goals. Section 4 describes our proposed approach, followed by the detailed evaluation of its security and performance in Section 5. Finally, we conclude this work in Section 6.

2 Related Work

2.1 Blockchain-Based Solutions for IoT Security

There have been increasing interests in designing blockchain-based solutions for IoT applications, partly evidenced by the number of survey articles on the topic [4], [5], [6]. Blockchain was used in designing a new distributed access control system for IoT systems [7], the design of FairAccess (a decentralized pseudonym and privacy protection authorization management framework for IoT devices) [8], to facilitate data integrity verification without involving a third-party auditor [3], and to facilitate authorization delegation and access control in IoT systems [9]. BeeKeeper [10], a blockchain-based threshold IoT service system, was designed to support servers in processing IoT data by performing homomorphism calculations, without the risk that other users learn the data directly.

To mitigate the dependency on third parties and reduce the cost of communication, Malik *et al.* [11] proposed an authentication framework based on blockchain for vehicular networks, and evaluated the performance of their framework using Omnet++ simulations. DeCusatis *et al.* [12] utilized the BlackRidge technology endpoint on a Windows host to implement a new identity management method for cloud-based blockchain applications. To avoid collusion and single point failure of centralized servers, Wu *et al.* [13] proposed an out-of-band two-factor authentication scheme for IoT devices using blockchain, and Miettinen *et al.* [14] proposed an IoT Sentinel Demo to identify and filter vulnerable devices.

Since the efficiency of transactions is also an impotent requirement during sharing information, especially in time and delay sensitive applications, such as agriculture, transport, agriculture, and transportation [15], there have been attempts to utilize smart contracts in different settings [16], [17], [18]. This is partly because smart contracts can support automated programming execution in a distributed manner (e.g., execution of transactions without a third-party), while ensuring traceability, efficiency and immutability. Smart contracts can run on virtual machines (VMs), e.g., Ethereum VMs (EVMs). Grishchenko, Maffei, and Schneidewind [19], for example, proposed the first complete small step semantics of EVM bytecode and a formalized proof assistant. Zhou *et al.* [20] built a static analysis tool, SASC, to generate the topology of the call relationships and uncover the potential logical risks. Due to the popularity of smart contracts in real-world applications, there have also been efforts in designing techniques, such as those using artificial intelligence (AI) to detect vulnerability in smart contracts [21], [22], [23], [24], [25].

Given the increasing emphasis on zero trust networked environments (e.g., all network traffic is considered untrustworthy, irrespective of the source), a number of security solutions, including those based on blockchain, have been designed to operate on such settings. For example, Lee *et al.* [26] suggested applying risk adaptable access control (RAcAC) to zero trust networks, as well as proposing a management framework to evaluate risks according to dynamically changing contexts. In a separate work, Vanickis *et al.* [27] designed an enforcement framework to tackle access control challenges in zero trust networks, and proposed a mechanism to map firewall rules to specific corresponding firewall syntax. To prove the viability of their framework, they evaluated a proof-of-concept implementation of their proposed mechanism. Samaniego *et al.* [28] provided a middleware based on blockchain in the process of zero trust hierarchical mining, designed to validate the infrastructure and transactions respectively at different trust levels. Another segmentation framework based on risk for zero trust IoT systems is that of Dhar *et al.* [29].

A number of researchers have proposed to utilize the blockchain to avoid the reliance on third parties for sharing information, since such an approach can potentially resist common attacks such as single point of failure and collusion attacks. Kim *et al.* [30] proposed an information-sharing scheme based on public blockchain to improve the security of self-driving. They also use cryptography and some protocols to realize privacy protection. Besides, Wang *et al.* [31] combined the structure of blockchain and algorithms of consensus to implement a decentralized sharing model for Government information resources (GIR). Some experiments results show that their scheme is safer and more efficient than traditional schemes. To tackle the challenge of integrating blockchain into the Mobile-edge computing

(MEC) system under the limited channel conditions and load, Liu *et al.* [32] proposed a secure data sharing framework for MEC system through an asynchronous learning approach. They also proposed an adaptive mechanism for privacy-preserving, with lower average throughput and consumption. Uploading and collecting plaintext data to the open cloud may lead to malicious tampering, which cannot meet the integrity and confidentiality of sharing information. Thus, to tackle above challenges, Ma *et al.* [33] proposed an encryption algorithm based on attributes through blockchain, which can implement accessing different data based on the attribute information.

Although these schemes consider utilizing blockchain, most of these schemes are not entirely decentralized. Besides, the fairness and privacy of participants (i.e., personal information and geographical positions) may be ignored in the sharing protocols. Thus, we propose a blockchain-enabled information-sharing protocol designed to operate in a zero-trust IoT, which can meet the requirement with fairness during the process of identity authentication and realize sharing process independently without relying on a TTP or compromising personal privacy.

2.2 Universally Composability Security Framework

The concept of universally composable (UC) security was first proposed by Canetti *et al.* [34] in 2001, which has also been applied in public-key encryption, signature, zero-knowledge, and identity authentication. In 2007, Canetti *et al.* [35] extended the definition of UC security to allow the existence of rogue protocols and to support non-repudiation. The proposed alternative assumptions and protocols are impossible to realize zero knowledge in standard reference string models. In the context of the UC security framework, Canetti *et al.* [36] provided minimum formalization of the 'ideal certification authority'. However, they explained how one could authenticate the communications by guaranteeing each party is analyzed independently in a modular and cryptographic way. Gajek *et al.* [37] proposed a security analysis of the TLS protocol, for example, by evaluating the critical exchange process through TLS handshake under a universal composable security framework. They also successfully emulated the communication sessions by transmitting messages at the TLS record layer. To achieve forward-secure anonymity, authenticity, and availability in the UC model, Tri *et al.* [38] utilized the UC framework for the protocols of RFID authentication and then proposed a novel, lightweight and practical protocol with a pseudo-random bit generator for crucial exchange and anonymous authentication. Peikert *et al.* [39] presented UC non-adaptive oblivious transfer protocols for the 1-out-of-2 variant. To address the challenge of extending to the adaptive k-out-of-N setting while also ensuring UC security, Green *et al.* [40] proposed a UC-secure adaptive k-out-of-N OT protocol, which is secure under bilinear assumptions. In the analysis of OpenStack under the UC security framework, Hogan *et al.* [41] explained the associated security challenges and provided alternative mechanisms to address these challenges.

Two decades after the proposal of the universal composability framework, the latter remains widely used in the security literature. For example, the protocols proposed by Ma *et al.* [42] in 2020 and Hao *et al.* [43] in 2021 were proven secure in this model. Hence, our choice of the security model in this paper is also the universal composability framework.

3 Problem Formulation

3.1 System Model

The entities in our system include the participants, the blockchain, smart contracts, and verified nodes. Participants need to authenticate each other's identities through the blockchain before sharing information. To ensure fairness of the sharing, we utilize smart contracts to execute some pre-defined operations (e.g., calculating the reputation score and verifying the correctness of the received data). Once the voting mechanism is triggered, the verification nodes will participate in the voting and choose the party to support. Finally, the smart contract determines which party is cheating based on the total number of votes. The purpose of the existence of verifying nodes is to avoid cheating participants.

3.2 Attacker Model

In our proposed approach, the attacker is capable of carrying out man-in-the-middle (MiTM) attacks, replay attacks, single point of failure attacks, and collusion attacks. In addition, the attacker may be a misbehaving participant, in order to obtain illicit gains.

Participator Misbehaving. The participators in the system may be compromised in terms of presenting any misbehavior, e.g., fake data is shared. We hereby assume that the misbehavior mainly intends to obtain more profits (awards).

Collusion Attacks. Work together with another user to carry out a nefarious activity, say cheating.

Man-in-the-Middle (MiTM) Attacks. Intercept and tamper with data-in-transit with the aim of carrying out activities to deceive the message sender and/or receiver.

Replay Attacks. Intercept data-in-transit and use the intercepted message in a subsequent communication session to deceive the target victim.

Single Point of Failure. We assume that any central server may also be compromised, or misused by mistakes.

3.3 Design Goals

The design goals of our approach are as follow:

1) Mutual authentication: Under the environment of zero trust, to ensure the identity of sharing parties, we verify the authenticity of identities through blockchain. Only legitimate participants can share information.
2) Fairness: After the sharing information process concludes, both participants will obtain the requested information. Honest sharers will be encouraged. Misbehaviors will be punished. If an allegation occurs or disputes, the consensus mechanism can be used to investigate this behavior, and the penalty mechanism can be invoked to enforce specific predefined penalties.
3) Autonomy: Consensus mechanism and information sharing process is controlled by smart contracts,

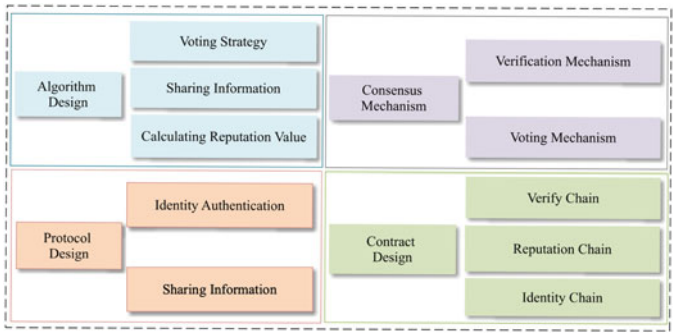

Fig. 1. High-level overview of our proposed scheme.

without relying on any third-party authority or centralized entity.

4) Anonymity: In the sharing information process, participants may worry about the leakage of their personal information (e.g., geographical positions and their sensitive or identifying information). Thus, blockchain is used to realize sharing real-time traffic information without exposing the participant's real identity.
5) Privacy Protection: The shared information should not leak the privacy of sharers, e.g., location information. To provide a general solution, the information may always be encrypted. It seems to be straightforward, but it is an extra advance in our solution.
6) Traceability: The used ID and public key will be stored in blockchain for tracing the information sharing history.

4 Our Proposed Scheme

Table 1 summarizes the notations used in this paper.

As shown in Fig. 1, there are four main modules in our scheme, namely: algorithm design, consensus mechanism, protocol design, and contract design.

Our scheme includes three main algorithms that utilize blockchain to implement the voting mechanism, the sharing of information, and reputation scores, respectively. To ensure fairness, we propose three mechanisms in the consensus mechanism module, namely: the verification mechanism, the voting mechanism, and proof of work. In addition, there are two main protocols in our scheme, namely: identity authentication and the sharing of information, and we analyze their security under the UC framework. We also remark that our method has three chains: the verify chain, the reputation chain, and the identity chain. These chains store different data in fixed structures to implement the various functions.

4.1 Protocol Design

For simplicity, let us assume that there are two participants, Alice and Bob, who wish to share some data in real-time (e.g., real-time traffic information) – see Fig. 2.

1) Alice uploads the value of Hash ($PK_A||ID_A||Timestamp$) to the blockchain. We assume the timestamp is sufficient to fetch the proper PKA. Otherwise, we can include a location information in the hash function for distinguishing multiple public keys, where the location information is pre-processed as a unit without concerning the leakage of location privacy.
2) Bob uploads the value of Hash ($PK_B||ID_B||Timestamp$) to the blockchain.
3) Bob generates Q and hashes it to acquire the value of H(Q). Then, Bob uses Alice's public key PK_A to encrypt K_B and Q, before sending $\{ID_B||K_B||Q||Timestamp)\}$ to Alice through PK_A's encryption.
4) Alice uses her SK_A to obtain the plaintext $\{ID_B||K_B||Q||Timestamp)\}$, and calculates the value of Hash($PK_B||ID_B$). Next, Alice checks whether the value of Hash($PK_B||ID_B$) equates the hash value on the blockchain.
5) After confirming Bob's ID through the blockchain, Alice generates a random number R, then hashes it to get H(R). Then, Alice uses K_B to encrypt the

TABLE 1
Summary of Notations

Enc	Encryption function
Dec	Decryption function
ID_A	Alice's identity
ID_B	Bob's identity
K_B	Bob's symmetric key
K_A	Alice's symmetric key
PK_B	Bob's public key
PK_A	Alice's public key
SK_B	Bob's private key
SK_A	Alice's private key
$DATA_A$	The shared information from Alice
$DATA_B$	The shared information from Bob
R	Random number generated by random number generator
Q	Another random number generated by random number generator
t	Time Slot
Th	Threshold
SR(t)	Sum reputation value
R(t)	Current reputation value
Z	External environment machine
SA	Ideal simulator
||	Connection symbol
F_{shr}	Information sharing ideal function
F_{BC}	Identity authentication ideal function

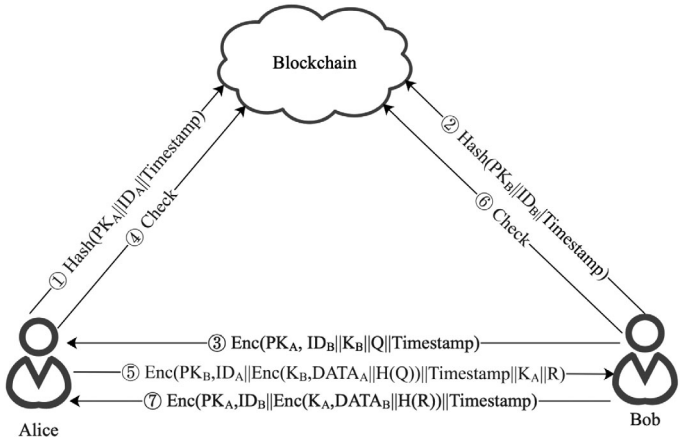

Fig. 2. The process of sharing information.

shared information $DATA_A$, and sends $\{ID_A||Enc_{K_B}(DATA_A||H(Q))||Timestamp||K_A||R\}$ to Bob through PK_B's encryption.

6) Bob uses SK_B to obtain the plaintext $\{ID_A||Enc_{K_B}(DATA_A||H(Q))||Timestamp||K_A||R\}$, and calculates the value of Hash($PK_A||ID_A$). Next, Bob checks whether the value of Hash($PK_A||ID_A$) is equal to the hash value on the blockchain.
7) After confirming Alice's ID through the blockchain and checking the value of H(Q). If the results are true, Bob uses K_A to encrypt the shared information $DATA_B$ and H(R), and sends $\{ID_B||Enc_{K_A}(DATA_B||H(R))||Timestamp\}$ to Alice; otherwise, Bob triggers the consensus mechanism.
8) Alice uses SK_A decryption to get $\{ID_B||Enc_{K_A}(DATA_B||H(R))||Timestamp\}$. After confirming Bob's ID through the blockchain and checking the value of H (R), if the results are true, the process of sharing information ends; otherwise, Alice triggers the consensus mechanism.

In order to analyze our scheme under the universally composable framework, we set F_{shr} and F_{BC} as ideal functionalities. Among them, F_{shr} represents information sharing and F_{BC} represents identity authentication. Besides, π_{shr} represents the protocols of sharing information. In our sharing information protocol π_{shr}, Z can capture the value and information of input and output of protocols. We set two sharing participants as $\{P_i \mid i = 1, 2, \ldots\}$ and $\{S_j \mid j = 1, 2, \ldots\}$, and the corresponding dummy participants as $\{P'_i \mid i = 1, 2, \ldots\}$ and $\{S'_j \mid j = 1, 2, \ldots\}$.

The functions of F_{shr} are elaborated as follows:

1) Z sends (initiative, sharing) to F_{shr}, it sets sharing tags to *shr*, F_{shr} sends (initialize, P_i) and (start, S_j) to F_{BC}, and receives the feedback of P_i and S_j from F_{BC}.
2) P_i sends (start, sharing, P_i) to F_{shr}, F_{shr} creates unique sharing label *sshr* and writes down (start, sshr, P_i), before sending to SA.
3) S_j sends (start, sharing, S_j) to F_{shr}, if S_j is occupied, this message is automatically ignored by F_{shr}; otherwise, F_{shr} creates unique sharing label *sshr*, then writes down (start, sshr, S_j) and sends to SA.
4) SA sends (start, sshr, S_j) and (start, sshr, P_i) to F_{shr}, F_{shr} deletes them and adds a new label (sharing, P_i, S_j, Timestamp), then generates a symmetric key pair K_B and calculates $C_1 = Enc_{PK_A}(ID_B||K_B||Q||Timestamp)$, then sends (ask, C_1, sshr, P_i, S_j) to P_i and SA.
5) SA sends (ask, C_1, sshr, P_i, S_j) to P_i, P_i calculates (Dec, C_1, sshr, P_i, S_j), then sends (checking, C_1, P_i, S_j, Timestamp) to F_{BC}. After receiving (accept, sharing, C_1, P_i, S_j, Time), F_{shr} generates a symmetric key pair K_A and calculates $C_2 = Enc_{PK_B}(ID_A \,||\, Enc_{K_B}(DATA_A \,||\, \text{H(Q)}) \,||\, \text{Timestamp} \,||\, K_A \,||\, \text{R})$, then sends (res, C_2, sshr, P_i, S_j) to S_j and S.
6) SA sends (res, C_2, sshr, P_i, S_j) to S_j, S_j calculates (Dec, C_2, sshr, P_i, S_j), then sends (checking, C_2, P_i, S_j, Timestamp) to F_{BC}. After receiving (accept, sharing, C_2, P_i, S_j, Timestamp) from F_{BC}, F_{shr} calculates $C_3 = Enc_{PK_A}(ID_B||Enc_{K_A}(DATA_B||H(R))||Timestamp)$, then sends (res, C_3, sshr, P_i, S_j) to P_i and SA.
7) After receiving (res, C_3, sshr, P_i, S_j) from SA, P_i calculates (Dec, C_3, sshr, P_i, S_j), then sends (checking, C_3, P_i, S_j, Timestamp) to F_{BL}. After receiving (accept, sharing, C_3, P_i, S_j, Timestamp) from F_{BC}, F_{shr} sends (succ, sharing, P_i, S_j) to P_i, S_j and SA.

Algorithm 1. Sharing Information Through Smart Contract

Input: *DepositA*, *DepositB*, PK_A, PK_B, T_1, T_2
Output: $Result_{Trans}$, *TimeStart*
Alice uploads *DepositA* and $H_A = H(ID_A)$ to the smart contract;
Bob uploads *DepositB* and $H_B = H(ID_B)$ the to smart contract;
$T_1 = T_2 = 0$;
if *The amount of DepositA and DepositB is right* **then**
  Bob generates Q and calculates H_Q = H(Q);
  Bob calculates $Enc_{PK_A}(ID_B||K_B||Q||Timestamp)$ and sends it to Alice;
  Alice decrypts and gets ID_B and Q, and then checks the result;
  if $H(ID_B) = H_B$ **then**
    Alice generates R and calculates H_R = H(R);
    Alice calculates $Enc_{K_B}(DATA_A||H(Q))$ and uploads it to the blockchain;
    Alice calculates $Enc_{PK_B}(ID_A||Enc_{K_B}(DATA_A||H(Q))||Timestamp||K_A||R)$ and sends it to Bob;
    Bob decrypts and gets ID_A, H(Q) and $DATA_A$, and then check this results;
    if $DATA_A$ *is right* **then**
      Trigger the receive mechanism;
      $T_1 = 1$;
    else
      Trigger the confirm verification mechanism;
    if $H(Q) = H_Q$ *and* $H(ID_A) = H_A$ **then**
      Bob calculates $Enc_{K_B}(DATA_B||H(R))$ and uploads it to blockchain;
      Bob calculates $Enc_{PK_A}(ID_B||Enc_{K_B}(DATA_B||H(R))\,||\,\text{Timestamp}\,||\,K_B)$ and send it to Alice;
      Alice decrypts and gets ID_B and H(R), and then check the result;
    if $H(R) = H_R$ *and* $H(ID_B) = H_B$ **then**
      Alice triggers the confirm receipt mechanism;
      $T_2 = 1$;
    else
      Alice triggers the confirm verification mechanism;
      if $T_1 = T_2 = 1$ **then**
      Result Trans = true;
    else
      Result Trans = false;

The functions of F_{BC} are elaborated as follows:

1) F_{shr} sends (checking, C_1, P_i, S_j, Timestamp) to F_{BC}, F_{BC} checks whether $H_A \epsilon Blockchain$ and Hash(ID_A) = H_A, then F_{BC} creates (accept, sharing, C_1, P_i, S_j, Timestamp) to F_{shr} and SA.
2) F_{shr} sends receiving (checking, C_2, P_i, S_j, Timestamp) to F_{BC}, F_{BC} checks whether $H_B \epsilon Blockchain$, Hash(ID_B) = H_B and whether the hash value of Q is right, then F_{BC} creates (accept, sharing, C_2, P_i, S_j, Timestamp) to F_{shr} and SA.

3) F_{shr} sends (checking, C_3, P_i, S_j, Timestamp) to F_{BC}, F_{BC} checks whether $H_A \epsilon Blockchain$, Hash(ID_A) = H_A and whether the hash value of R is right, then F_{BC} creates (accept, sharing, C_3, P_i, S_j, Timestamp) to F_{shr} and SA.

Algorithm 2. Smart Contract for Calculating Reputation Value

Output: R(t), SR
Set *Th* = 100;
Set t = 1;
Set R(0) = r(t) = 10;
α = Math.random() ;
/* Randomly select a number greater than 0 and less than 1 */
voting(); /* Determine whether the user provides false information */
if *voting() = 1* **then**
  r(t)++;
else
  r(t)–;
if *t <Th and* **then**
  R(t) = α* R(t-1) + r(t);
  t++;
  SR = $\sum_{t=0}^{Th} R(t)$;
else
  $\bar{RSU}_i$ compares participants' SR(t) and chooses the highest SR(t);
Package the vehicle's ID, signature, SR(t) and Transactions List, then upload to blockchain;

4.2 Key Algorithms

Next, we will describe the three main algorithms the underpin the smart contracts in our scheme, namely: the sharing processing (see Algorithm 1), the reputation value and sum reputation value computation (see Algorithm 2), and the voting mechanism (see Algorithm 3). In Algorithm 2, we assume that *Th* is 100 although in practice, the value of threshold *Th* can be set by systems or selected by participants. It should be noted that we set $0 < \alpha < 1$ to reduce the contribution of *r(t-1)* and increase the contribution of *r(t)*. At the beginning, every participant is assigned the same initiate value of *r(0)*. When a user is determined to be misbehaving (e.g., cheating), *r(t)* will be decremented by 1. When *r(t)* is 0, the corresponding user will be denied access to future transactions.

4.3 Consensus Design

Since we assume no participant can be trusted in the zero trust environment, we need to introduce mechanisms to detect and penalize misbehaving or malicious participants. However, how do we determine whether a piece of information to be true or false? We propose a verification mechanism that builds on blockchain consensus to determine the authenticity of the message. Let $DATA_A$ and $DATA_B$ denote the received data, and the following scenario:

- Alice sends $DATA_A$ to Bob;
- Bob provides the information that is inconsistent with the facts described by $DATA_A$ he received;
- Other neighboring blockchain nodes vote for either Alice or Bob, by sending Alice or Bob their vote;
- The smart contract counts the total number of Alice and Bob, and announces the winner. The individual with the less votes loses and his/her deposit will be deducted. The deposit amount should be set at a rate that can effectively serve as a deterrent for misbehavior or malicious behavior (e.g., false accusation or injecting malicious / fabricated message).

Algorithm 3. Smart Contract for Voting Mechanism

Parameters:
Voter.Weight; /* Number of votes owned */
Voter.VoteA; /* The total number of Alice's votes */
Voter.VoteB; /* The total number of Bob's votes */
Voter.Right /* Whether this participant has voting rights */
Voter.delegate; /* The address which user the voter desires to support */
Proposal.voteCount;
1. **Tectonic stage:**
/* We assume Bob claims that Alice cheated him. Then, Bob uploads materials and calls smart contract as a chairperson. */
Initialization parameters:
NumProposal; /* The total number of participants */
Chairperson; /* The person who initiated the vote */
weight = 1; /* Who has right to vote */
Chairperson enters the address of participants who is given the right to vote;
2. **Voting stage:**
Voter.delegate(); /* Voter enters the supported address of Alice or Bob */
if $Voter.weight \geq 0$ **then**
  /* Judge whether this participant has voting rights */
  Voter.Right =1 ;
else
  Voter.Right =0;
if $NumProposal \geq 0$ **then**
  Total votes for Alice or Bob + = Weight of voters;
  Voter.Right = 0; /* This participant does not have voting rights */
  Weight for Alice or Bob + = Weight for voters;
  NumProposal–;
3. **Counting stage:**
if $count(VoteA) \geq count(VoteB)$ **then**
  Deducting Bob's deposite;
  /* The deducted deposit will be assigned to Alice and all participants who had voted. */
else
  Deducting Alice's deposit;
  /* The deducted deposit will be assigned to Bob and all participants who had voted. */

Next, we will describe the consensus mechanism in the use case of smart vehicles shown in Fig. 3. Our plan consists of two parts: the authentication mechanism and the consensus mechanism. When either party A or B finds that the received message is wrong, he/she can choose to trigger the consensus mechanism to verify the accuracy of the message. This model has five entities: roadside units (RSUs), smart contracts, vehicles, shared information files, and blockchain. Among them, vehicle A and vehicle B desire to share real-time road information. The information can be packaged

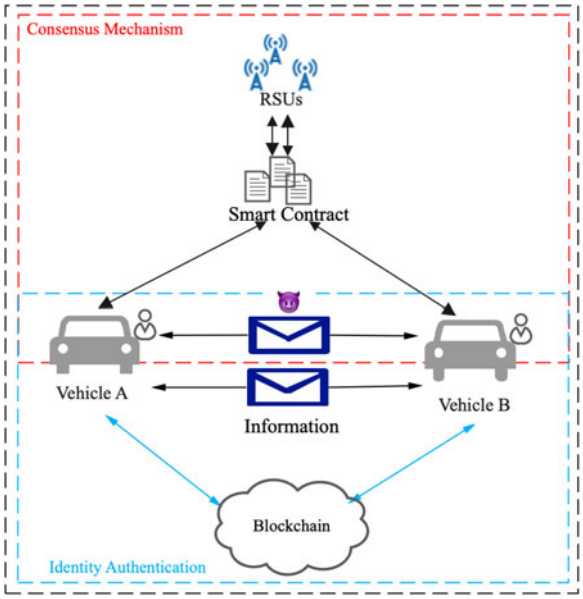

Fig. 3. Information sharing process for smart vehicles.

TABLE 2
The Structure of Identity Chain

Attribute	Definition
ID_i	The ID of participant i
PK_i	The public key of participant i
$HASH(ID_i\|\|PK_i)$	The hash value of ID and public key of participant i

TABLE 3
The Structure of Reputation Chain

Attribute	Definition
Th	Threshold value of packaging a new block on reputation chain
SR_i	User i with the highest sum reputation value
TL	Transaction lists

and sent to the receivers in files. RSUs can collect some voting data through smart contracts. When two vehicles need to share information, they must complete the identity authentication through the blockchain. After confirming the participant identities, they can share information (e.g., real-time traffic information). However, we also need to ensure the authenticity of the transmitted/received information. Thus, we propose the following consensus mechanism through RSUs and smart contracts.

First, we set a threshold *Th*, when the number of transactions is *Th*, and the RSU packages the block that has the highest sum reputation value (sr). The specific block structure is shown in Fig. 4. Except for the general value, we record the vehicle's ID and signature to ensure non-repudiation (i.e., participants cannot deny their transactions). We also introduce the concept of bonus, which will be assigned to the chosen participants and miners. This mechanism aims to encourage participants to share information without misbehavior. In other words, participants will send the correct information as much as possible to increase their reputation values by avoiding making suspicious behaviors.

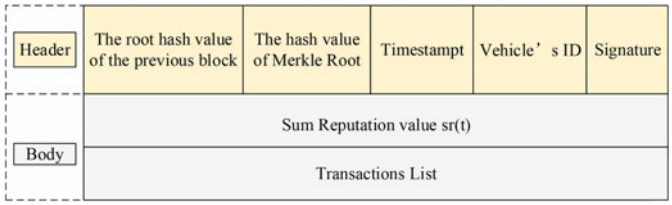

Fig. 4. The block's structure.

TABLE 4
The Structure of Verify Chain

Attribute	Definition
ID_A	Verified participant A in this transcation
ID_B	Another verified participant A in this transcation
Sum_A	The total number of votes of A
Sum_B	The total number of votes of B
Win_i	The honest participant i

According to the value of SR, the user will be given a corresponding credit rating (e.g., either on a scale of 1 to 10, or ratings such as excellent, good or bad). Users having a higher rating will be given priority to participate in future sharing process.

4.4 Smart Contract Design

Next, we describe the functions of smart contracts, and there are three key chains (i.e., identity chain, reputation chain and verify chain). The identity chain stores the hash values of the user's PK and ID, which can be used to ensure the participant's identity, and the corresponding defined structure of identity chain is shown in Table 2. Specifically, after acquiring both ID_A and PK_A, another user, say Bob, can connect these two strings and calculate a hash value, then compare this value with the corresponding hash value stored on the identity chain to verify A's identity.

Next, we describe the structure of the reputation chain (see Table 3). In this chain, the smart contract records the number of transaction lists and calculates the sum reputation value of participant *i*, until the number reaches the set threshold *Th*. The participants with the highest sum reputation value will be recorded in the reputation chain.

In order to guarantee fairness of the information sharing, the verify chain is proposed to record the voting result and participants nodes (see also Table 4). In this chain, smart contracts need to record the total number of votes of two participants respectively and compare the results, prior to declaring the winner based on the voting results.

4.5 Discussions

We consider how to speed up the verification while ensuring accuracy, so we propose reducing the number of verification nodes. Our scheme adopts an incentive mechanism, which uses part of the deposit to reward the participated nodes in the verification. Reducing the number of verification nodes has become an advanced scheme to reduce the

bonus. It brings a problem - ensuring fairness and accuracy while reducing the number of nodes. If the verification nodes can be randomly selected online, the above issue can be addressed, and then some malicious attacks can also be avoided. In future work, we consider improving our scheme from the following aspects: reducing communication, calculation, time delay and improving security (including fault tolerance, anti node compromise rate).

5 Security and Performance Evaluations

Next, we will evaluate the security and performance of our scheme (see Sections 5.1 and 5.3).

5.1 Informal Security Analysis

Traceability and Tamper-Resistant. Based on the characteristics of blockchain and smart contracts, as long as the majority of nodes do not reach an agreement, our scheme can also achieve the requirement of traceability and tamper-resistant.

Confidentiality and Integrity. Due to the tamper-resistant and the traceability of blockchain, our scheme can prevent reading without authentic writing and execution. Besides, when the receiver discovers that the information has been tampered with (e.g., the message does not match the fact), he can trigger the authentication and consensus mechanism. According to the traceability of our scheme, the malicious user's reputation value will be reduced and will receive the corresponding penalty.

Privacy. Sensitive and personal information (e.g., real-time geographical positions and vehicle IDs) can be preserved as long as the majority of the nodes are not compromised due to the anonymity of blockchain and smart contracts.

MiTM Attack Resilience. We assume that there is an attacker, say Eve, seeking to disrupt the information sharing between Alice and Bob without exposing herself. Specifically, Bob use PK_A to encrypt K_B and Q, and then sends $\{ID_B||K_B||Q||Timestamp\}$ to Alice. Eve attempts to intercept the communication and parse the information. However, Eve cannot get K_B since only having the private key SK_A will allow her to decrypt this ciphertext. Similarly, if getting $Enc_{PK_B}(ID_A$ || Enc_{K_B} ($DATA_A$ || H(Q)) || Timestamp ||K_A || R) without SK_B, will not reveal K_A and $DATA_A$. Thus, our scheme can resist MiTM attacks.

Replay Attack Resilience. We add the value of current time in the shared information, including $Enc_{PK_A}(ID_B$ || K_B || Q || Timestamp), $Enc_{PK_B}(ID_A$ || Enc_{K_B} ($DATA_A$ || H (Q)) || Timestamp || $K_A||R$), and $Enc_{PK_A}(ID_B$ || Enc_{K_B} ($DATA_B$ || H(R)) || Timestamp || K_B). Suppose attackers attempt to reply to the previously shared information, since the current time and random number exist, it is easy for the receivers to find this attack of the mismatches between the current time and the information.

Single Point of Failure and Collusion Attack Resilience. Our scheme proposes an information-sharing protocol in zero-trust IoT through smart contracts. The participants can autonomously complete the sharing process independently without a third trusted party, thus avoiding a single point of failure and collusion attacks.

Fairness. We have proposed a verification mechanism and a consensus mechanism to ensure the authenticity of the received information as well as detect and filter false information. As for the verification mechanism, if a participant declares that the user sent some false information, the verification mechanism will be triggered, and other neighboring nodes can help identify misbehaving participants. To ensure the fairness of shared information, we have also proposed a consensus mechanism with reputation value. In this mechanism, we set a threshold and use a combination of RSUs and smart contracts to ensure the effectiveness of the consensus mechanism. When the threshold reaches a specific number, the account will be prohibited from trading. Therefore, our scheme can satisfy the fairness of sharing information.

We also consider the probability of 51% attack or collusion attack. We set three parameters: a - the probability of one person defecting, m - the number of mutineers, and n - total number of participators in the consensus mechanism. It is worth noting that we assume the probability of one person defecting is less than $1/2$. Next, we calculate the probability of 51% attack.

$$\begin{aligned} P[51\%\ attack] &= P[m/n \geq 1/2] \\ &= max(P[m = n/2+1], P[m = n/2+2], \\ &\quad \ldots, P[m = n]) \\ &= max(C_n^{n/2+1} * a^{n/2+1} * (1-a)^{n/2-1}, \ldots, \\ &\quad C_n^n * a^n * (1-a)^0) \\ &\leq a^n. \end{aligned}$$

Since a is less than or equal to $1/2$, the result is small when n is large enough.

$$P[51\%\ attack] \leq negl(n).$$

According to the above analysis, we can conclude that the probability of 51% attack in our scheme is negligible.

5.2 Security Proof

Next, we will analyze our blockchain-enabled solution under the universal composability security framework.

Theorem 1. *Suppose that A is a real adversary and SA is an ideal simulator. In this case, indistinguishability means that Z cannot distinguish between the environment where π_{shr} and A interact with the environment where F_{shr}, F_{BC} and S interact. That is, π_{shr} can implement the ideal function F_{shr} safely in F_{BC} - hybrid model.*

Proof. *Proof of indistinguishability*: First, we define some possible attacks simulated by SA as follows:

1) If F_{shr} sends (start, sshr, P_i) to SA, it calculates a new sshr' and transfers (start, sshr', P_i) to P.
2) If S_j sends (ask, C_1, sshr, P_i, S_j) to SA, it calculates a new $C_1 = Enc_{PK_A}(ID_B||K_B||Q||Timestamp)$ and sends (ask, C_1, sshr, P_i, S_j) to F_{shr}.
3) If P_i sends (res, C_2, sshr, P_i, S_j) to SA, it calculates a new $C_2 = Enc_{PK_B}(ID_A$ || Enc_{K_B} ($DATA_A$ || H (Q)) || Timestamp ||K_A || R) sends (res, C_2, sshr, P_i, S_j) to F_{shr}.

TABLE 5
Experimental Environment Configuration

Attributes	Version numbers
Operating system	Windows 10
Programming language	Python, Solidity
Development environment	Python3.7, Truffle, Ganache, Web3

4) If S_j sends (res, C_3, sshr, P_i, S_j) to SA, it generates a new $C_3 = Enc_{PK_A}(ID_B||Enc_{K_A}(DATA_B||\ H(R))||\ Timestamp)$, sends (res, C_3, sshr, P_i, S_j) to F_{shr}.
5) If F_{BC} sends (accept, P_i, S_j, Timestamp) to SA, SA transfers the signal to P and justifies whether $H_B \epsilon Blockchain$, Hash(ID_B) = H_B and the hash value of Q is right, then sends (accept, P_i, S_j, Timestamp) to F_{shr}.

Then, we define an event P that SA can simulate the above attacks successfully, for example, SA can forge new C_1', C_2', C_3'. The calculation of these three values includes hash functions and random functions. Since it is challenging to construct a PPT algorithm to find a collision of hash functions or random functions, the successful rate of event P is negligible. That is, this simulation is reasonable, π_{shr} can implement the ideal function F_{shr} safely in F_{BC} - hybrid model. □

5.3 Performance Analysis

In our experiments, we use Ganache to develop blockchain applications. We use the Web3 module to write a Python program to interact with the nodes in Ganache blockchain. That is, Web3 is the interface between users and blockchain nodes. Truffle is used to write smart contracts, which can be deployed and invoked on Ganache. The system environment configuration is shown in Table 5. The results of time consumption are demonstrated in Fig. 5. It is worth noting that the three curves in different colors correspond to different steps of information sharing in Fig. 2. We can find that the average execution time of the three critical steps in sharing information protocol is 0.059s, 0.060s, and 0.032s.

Alice and Bob will store their identity information on blockchain in sharing information. The function of the smart

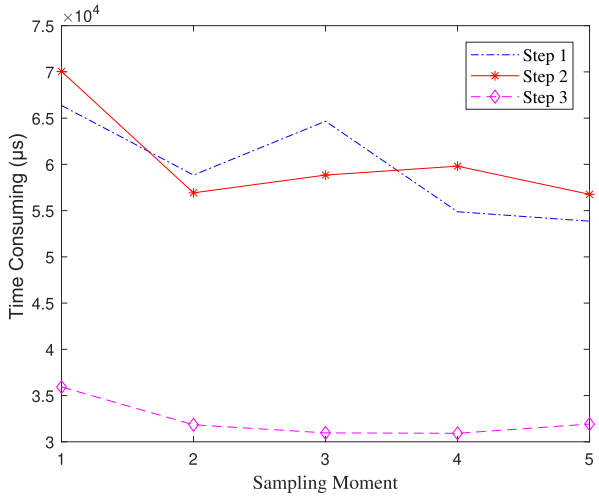

Fig. 5. The time consuming of sharing information.

TABLE 6
The Value of Transaction and Execution Cost

Four main steps	The value of transaction cost	The value of execution cost
Deployment contract	1111274 gas	834962 gas
Voting to users once	83080 gas	61616 gas
Giving voting rights once	44076 gas	21396 gas
Call winner() once	26293 gas	5021 gas

contract is to store the hash value identity data. For example, Alice wants to upload her information on blockchain, and the input is the hash value of the connection of Alice's public key and identity information. Once Alice calls the identity storage smart contract, the identity information will be stored on blockchain. To conclude, the smart contract used in sharing information is to store identity information and shared data. In solidity, the hash value returns bytes32 data, and the time is uint256 type of data. As for ID, it is a type of bytes32 data, which is the conversion of string. The public key is in the form of strings. As for the reputation calculation smart contract, its function is to calculate the reputation value of the users on blockchain. The input of the smart contract is voter's address, voter's weight value, and the number of voters. The output is the reputation values of users. In solidity, we define each voter as structured data, in terms of voter's address, voter's vote value, and voter's rights. Each vote is a uint8 data. Besides, there is a uint8 data named 'voting' to record whether the voter has voted or not.

Besides, we test the value of transaction and execution cost from four main steps in the voting process respectively, including deployment contract, voting to users once, giving voting rights once, call winner() once, as shown in Table 6. Each raw represents the gas consumed by executing each step once. In addition, we compare our scheme with some related references, including integrity, privacy protection, identity, authentication, fairness, and time complexity, which is shown in Table 7, to prove the advantages of our scheme.

5.4 More Applications

In this section, we will introduce potential applications at the following three levels. We remark that the protocol layer comprises the storage layer and the network layer, where these two layers are independent.

Protocol Layer. The protocol layer refers to the underlying technology. This level is usually a complete blockchain product, similar to our computer's operating system. It maintains network nodes and only provides API for calling. The interface functions offered also very simple, basically all of which are network programming, distributed algorithms, encrypted signatures, and data storage technologies. The most prominent feature of blockchain in industrial applications is security. Once a hacker breaks through the firewall, the centralized storage database behind it is naked, allowing the hacker to read all the data stored in the database. In the process of realizing point-to-point networks, distributed algorithms and encrypted signatures are also used. There are also separate implementations of peer-to-

TABLE 7
Scheme Comparison

	Integrity	Privacy Protection	Identity Authentication	Fairness	Time Complexity
Kim [30]	✓	✓	×	×	O(n)
Wang [31]	✓	Not mentioned	✓	×	$O(n^2)$
Liu [32]	✓	✓	✓	×	$O(n^2)$
Ma [33]	✓	✓	✓	×	$O(n^2)$
Our scheme	✓	✓	✓	✓	O(n)

peer networks, independent logic such as variable search, data transmission, and verification, and put consensus algorithms, encryption signatures, data storage, and other operations together to form the core layer.

Extension Layer. The extension layer can be understood as the device driver we use to make blockchain products more practical. It can be an interactive platform or a comprehensive implementation, such as writing smart contracts. The academic language is called 'programmable dynamic contracts'. The intelligence is mainly reflected in execution intelligence, and the contract will automatically take effect when a specific condition is reached. The deployed smart contract has decentralized and tamper-proof modification, such as automatic transfer of Bitcoin, automatic payment, and smart contract mechanism for purchasing and invoice business in industrial applications. Some voting and auction mechanisms can be implemented on smart contracts. From this perspective, the blockchain can be used to develop any product, not just in the financial industry. In the future, in the industry, with the improvement of the underlying protocols, any information that requires confirmation, credit investigation and traceability can be realized with the help of blockchain.

Application Layer. The application layer of the blockchain computer software programs and mobile phone app applications are products that the public can directly use. We also introduce the information-sharing application in Internet-of-Things, which is shown in Fig. 6. Some camera nodes need to share some information, such as some product pass rate information, to adjust the material supply in time. For example, a video camera node on a factory's production line can learn and obtain the information of qualified rate of products and then share it with another factory's node to control the raw material supply of the product.

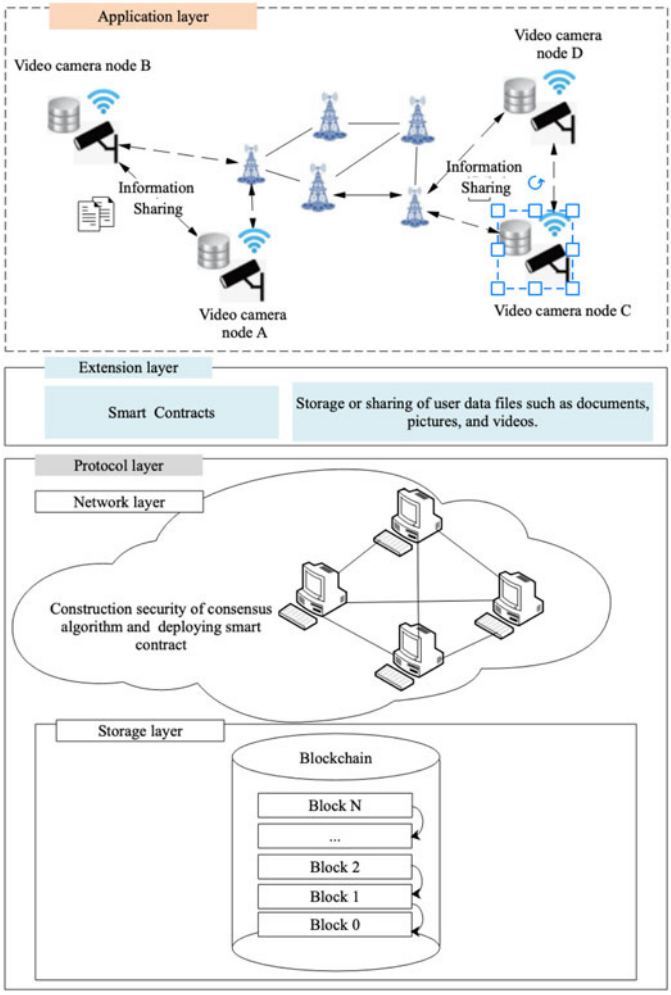

Fig. 6. Industrial applications.

6 Conclusion

We described our proposed blockchain-based scheme, designed to facilitate information sharing in a zero-trust IoT environment, together with anonymity traceability, temporary identity authentication, and data privacy protection. Specifically, our approach uses smart contracts as one of the key building blocks to avoid the reliance on third-party entities. We also introduced a detection mechanism to minimize the risk of false information dissemination and a penalty mechanism to enforce pre-defined penalties on misbehaving participants. We then proved the security of our scheme in the universally composable framework and evaluated its performance. One of the future research extensions is to investigate how to further reduce the communication delay and the computation overheads of the protocol. In addition, we intend to implement the enhanced scheme in a real-world environment to evaluate its security and performance.

Data Availability Statement

The [code] data used to support the findings of this study have been deposited in the [IEEE DATAPORT] repository ([10.21227/rtmq-t937]).

References

[1] B. Qianetal.,"Orchestrating the development lifecycle of machine learning-based IoT applications: A taxonomy and survey," *ACM Comput. Surv.*, vol. 53, no. 4, pp. 1–47, 2020.

[2] Y. Zhang and J. Wen,"The IoT electric business model: Using blockchain technology for the Internet of Things," *Peer-to-Peer Netw. Appl.*, vol. 10, no. 4, pp. 983–994, 2017.
[3] B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu, "Blockchain based data integrity service framework for IoT data," in *Proc. IEEE Int. Conf. Web Serv.*, 2017, pp. 468–475.
[4] L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, andY. Yang,"A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling," *ACM Comput. Surv.*, vol. 53, no. 1, pp. 1–32, 2020.
[5] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Integration of blockchain and cloud of things: Architecture, applications and challenges," *IEEE Commun. Surv. Tuts.*, vol. 22, no. 4, pp. 2521–2549, Oct.-Dec. 2020.
[6] N. Waheed, X. He, M. Ikram, M. Usman, S. S. Hashmi, andM. Usman,"Security and privacy in IoT using machine learning and blockchain: Threats and countermeasures," *ACM Comput. Surv.*, vol. 53, no. 6, pp. 1–37, 2020.
[7] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1184–1195, Apr. 2018.
[8] A. Ouaddah, A. Abou Elkalam, and A. A. Ouahman, "Towards a novel privacy-preserving access control model based on blockchain technology in IoT," in *Proc. Eur. MENA Cooperation Adv. Inf. Commun. Technol.*, 2017, pp. 523–533.
[9] G. Ali, N. Ahmad, Y. Cao, M. Asif, H. Cruickshank, and Q. E. Ali, "Blockchain based permission delegation and access control in Internet of Things (BACI)," *Comput. Secur.*, vol. 86, pp. 318–334, 2019.
[10] L. Zhou, L. Wang, Y. Sun, and P. Lv, "BeeKeeper: A blockchain-based IoT system with secure storage and homomorphic computation," *IEEE Access*, vol. 6, pp. 43 472–43 488, 2018.
[11] N. Malik, P. Nanda, A. Arora, X. He, and D. Puthal, "Blockchain based secured identity authentication and expeditious revocation framework for vehicular networks," in *Proc. 17th IEEE Int. Conf. Trust Secur. Privacy Comput. Commun./12th IEEE Int. Conf. Big Data Sci. Eng.*, 2018, pp. 674–679.
[12] C. DeCusatis, M. Zimmermann, and A. Sager, "Identity-based network security for commercial blockchain services," in *Proc. IEEE 8th Annu. Comput. Commun. Workshop Conf.*, 2018, pp. 474–477.
[13] L. Wu, X. Du, W. Wang, and B. Lin, "An out-of-band authentication scheme for Internet of Things using blockchain technology," in *Proc. Int. Conf. Comput. Netw. Commun.*, 2018, pp. 769–773.
[14] M. Miettinen *et al.*, "IoT sentinel demo: Automated device-type identification for security enforcement in IoT," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 2511–2514.
[15] M. Barika, S. Garg, A. Y. Zomaya, L. Wang, A. V. Moorsel, andR. Ranjan,"Orchestrating big data analysis workflows in the cloud: Research challenges, survey, and future directions," *ACM Comput. Surv.*, vol. 52, no. 5, pp. 1–41, 2019.
[16] M. Bartoletti and L. Pompianu, "An empirical analysis of smart contracts: Platforms, applications, and design patterns," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, 2017, pp. 494–509.
[17] P. Siano, G. De Marco, A. Rolán, and V. Loia, "A survey and evaluation of the potentials of distributed ledger technology for peer-to-peer transactive energy exchanges in local energy markets," *IEEE Syst. J.*, vol. 13, no. 3, pp. 3454–3466, Sep. 2019.
[18] Á. J. Varela-Vaca and A. M. R. Quintero,"Smart contract languages: A multivocal mapping study," *ACM Comput. Surv.*, vol. 54, no. 1, pp. 1–38, 2021.
[19] I. Grishchenko, M. Maffei, and C. Schneidewind, "A semantic framework for the security analysis of ethereum smart contracts," in *Proc. Int. Conf. Princ. Secur. Trust*, 2018, pp. 243–269.
[20] E. Zhou *et al.*, "Security assurance for smart contract," in *Proc. 9th IFIP Int. Conf. New Technol. Mobility Secur.*, 2018, pp. 1–5.
[21] B. Jiang, Y. Liu, and W. Chan, "ContractFuzzer: Fuzzing smart contracts for vulnerability detection," in *Proc. 33rd IEEE/ACM Int. Conf. Autom. Softw. Eng.*, 2018, pp. 259–269.
[22] A. Mavridou and A. Laszka, "Designing secure ethereum smart contracts: A finite state machine based approach," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, 2018, pp. 523–540.
[23] F. Wu, J. Wang, J. Liu, and W. Wang, "Vulnerability detection with deep learning," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2018, pp. 1298–1302.
[24] R. M. Parizi, A. Dehghantanha, K. R. Choo, and A. Singh, "Empirical vulnerability analysis of automated smart contracts security testing on blockchains," in *Proc. 28th Annu. Int. Conf. Comput. Sci. Softw. Eng.*, 2018, pp. 103–113.
[25] W. Wang, J. Song, G. Xu, Y. Li, H. Wang, and C. Su, "ContractWard: Automated vulnerability detection models for ethereum smart contracts," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1133–1144, Apr.-Jun. 2021.
[26] B. Lee, R. Vanickis, F. Rogelio, and P. Jacob, "Situational awareness based risk-adaptable access control in enterprise networks," in *Proc. 2nd Int. Conf. Internet Things Big Data Secur.*, 2017, pp. 400–405.
[27] R. Vanickis, P. Jacob, S. Dehghanzadeh, and B. Lee, "Access control policy enforcement for zero-trust-networking," in *Proc. 29th Ir. Signals Syst. Conf.*, 2018, pp. 1–6.
[28] M. Samaniego and R. Deters, "Zero-trust hierarchical management in IoT," in *Proc. IEEE Int. Congress Internet Things*, 2018, pp. 88–95.
[29] S. Dhar and I. Bose, "Securing IoT devices using zero trust and blockchain," *J. Organizational Comput. Electron. Commerce*, vol. 31, no. 1, pp. 18–34, Nov. 2020.
[30] K. Kim, T. Kim, and I. Y. Jung, "Blockchain-based information sharing between smart vehicles for safe driving," in *Proc. IEEE 91st Veh. Technol. Conf.*, 2020, pp. 1–2.
[31] L. Wang, W. Liu, and X. Han, "Blockchain-based government information resource sharing," in *Proc. IEEE 23rd Int. Conf. Parallel Distrib. Syst.*, 2017, pp. 804–809.
[32] L. Liu *et al.*, "Blockchain-enabled secure data sharing scheme in mobile-edge computing: An asynchronous advantage actor–critic learning approach," *IEEE Internet of Things J.*, vol. 8, no. 4, pp. 2342–2353, Feb. 2021.
[33] J. Ma, T. Li, J. Cui, Z. Ying, and J. Cheng, "Attribute-based secure announcement sharing among vehicles using blockchain," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10873–10883, Jul. 2021.
[34] R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," in *Proc. 42nd IEEE Symp. Found. Comput. Sci.*, 2001, pp. 136–145.
[35] R. Canetti, Y. Dodis, R. Pass, and S. Walfish, "Universally composable security with global setup," in *Proc. Theory Cryptogr. Conf.*, 2007, pp. 61–85.
[36] R. Canetti, "Universally composable signature, certification, and authentication," in *Proc. 17th IEEE Comput. Secur. Found. Workshop*, 2004, pp. 219–233.
[37] S. Gajek, M. Manulis, O. Pereira, A.-R. Sadeghi, and J. Schwenk, "Universally composable security analysis of TLS," in *Proc. Int. Conf. Provable Secur.*, 2008, pp. 313–327.
[38] T. Van Le, M. Burmester, and B. de Medeiros, "Universally composable and forward-secure RFID authentication and authenticated key exchange," in *Proc. 2nd ACM Symp. Inf. Comput. Commun. Secur.*, 2007, pp. 242–252.
[39] C. Peikert, V. Vaikuntanathan, and B. Waters, "A framework for efficient and composable oblivious transfer," in *Proc. Annu. Int. Cryptol. Conf.*, 2008, pp. 554–571.
[40] M. Green and S. Hohenberger, "Universally composable adaptive oblivious transfer," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2008, pp. 179–197.
[41] K. Hogan *et al.*, "On the universally composable security of openstack," in *Proc. IEEE Cybersecurity Develop.*, 2019, pp. 20–33.
[42] Z. Ma, J. Zhang, Y. Guo, Y. Liu, X. Liu, and W. He, "An efficient decentralized key management mechanism for VANET with blockchain," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5836–5849, Jun. 2020.
[43] X. Hao, W. Ren, K.-K. R. Choo, and N. Xiong, "A self-trading and authenticated-roaming scheme based on blockchain for smart grid," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 4097–4106, Jun. 2022.

Yizhi Liu received the bachelor's and master's degrees from Wuhan University, China. He is currently working toward the PhD degree with the School of Computer Science, China University of Geosciences, Wuhan, China. His main research interests include Big Data security, including Big Data of multiple source Geo-information, remote sensing and Geo-informatics, and security and privacy concerns.

Xiaohan Hao received the graduate degree from the School of Computer Science, China University of Geosciences, Wuhan, China, in 2020. She is currently working toward the PhD degree with the School of Software, University of Technology Sydney, Australia. Her research interests include Internet of Things, cryptography, and blockchain.

Wei Ren (Member, IEEE) received the PhD degree in computer science from the Huazhong University of Science and Technology, China. He is currently a full professor with the School of Computer Science, China University of Geosciences, Wuhan, China. He was with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, Illinois, in 2007 and 2008, School of Computer Science, University of Nevada Las Vegas, Las Vegas, Nevada, in 2006 and 2007, and Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong, in 2004 and 2005. He has authored or coauthored more than 100 refereed papers, one monograph, and four textbooks. He has obtained ten patents and five innovation awards. He is a distinguished member of China Computer Federation.

Ruoting Xiong received the graduate degree from the School of Computer Science, China University of Geosciences, Wuhan, China. She is currently working toward the master's degree with the School of Computer Science, Huazhong University of Science and Technology, China. Her research interests include access control and cryptography.

Tianqing Zhu (Member, IEEE) received the BEng and MEng degrees from Wuhan University, China, in 2000 and 2004, respectively, and the PhD degree in computer science from Deakin University, Australia, in 2014. She is currently a professor with the China University of Geosciences, Wuhan, China. Prior to that, she was a lecturer with the School of Information Technology, Deakin University, Australia. Her research interests include privacy preserving, AI security and privacy, and network security.

Kim-Kwang Raymond Choo (Senior Member, IEEE) received the PhD degree in information security from the Queensland University of Technology, Australia, in 2006. He currently holds the Cloud Technology Endowed professorship with The University of Texas at San Antonio, San Antonio, Texas. He is the founding co-editor-in-chief of *ACM Distributed Ledger Technologies: Research & Practice*, and founding chair of IEEE TEMS Technical Committee on Blockchain and Distributed Ledger Technologies. He is an ACM distinguished speaker and IEEE Computer Society distinguished visitor (2021–2023), a Web of Science's Highly Cited Researcher (Computer Science—2021, Cross-Field—2020), and the recipient of the 2019 IEEE Technical Committee on Scalable Computing Award for Excellence in Scalable Computing (Middle Career Researcher.

Geyong Min (Member, IEEE) received the BSc degree in computer science from the Huazhong University of Science and Technology, China, in 1995, and the PhD degree in computing science from the University of Glasgow, U.K., in 2003. He is currently a professor of high performance computing and networking with the Department of Computer Science, College of Engineering, Mathematics and Physical Sciences with the University of Exeter, U.K. His research interests include computer networks, wireless communications, parallel and distributed computing, ubiquitous computing, multimedia systems, modeling, and performance engineering.