Computers, Materials & Continua CMC, vol.65, no.3, pp.2365-2384, 2020

Data Secure Storage Mechanism of Sensor Networks Based on Blockchain

Jin Wang[1, 2], Wencheng Chen[1], Lei Wang[3, *], R. Simon Sherratt[4], Osama Alfarraj[5], and Amr Tolba[5, 6]

Abstract: As the number of sensor network application scenarios continues to grow, the security problems inherent in this approach have become obstacles that hinder its wide application. However, it has attracted increasing attention from industry and academia. The blockchain is based on a distributed network and has the characteristics of non-tampering and traceability of block data. It is thus naturally able to solve the security problems of the sensor networks. Accordingly, this paper first analyzes the security risks associated with data storage in the sensor networks, then proposes using blockchain technology to ensure that data storage in the sensor networks is secure. In the traditional blockchain, the data layer uses a Merkle hash tree to store data; however, the Merkle hash tree cannot provide non-member proof, which makes it unable to resist the attacks of malicious nodes in networks. To solve this problem, this paper utilizes a cryptographic accumulator rather than a Merkle hash tree to provide both member proof and non-member proof. Moreover, the number of elements in the existing accumulator is limited and unable to meet the blockchain's expansion requirements. This paper therefore proposes a new type of unbounded accumulator and provides its definition and security model. Finally, this paper constructs an unbounded accumulator scheme using bilinear pairs and analyzes its performance.

Keywords: Sensor networks, blockchain, unbounded accumulator, storage mechanism.

1 Introduction

The sensor network is generally regarded as the third revolution in the development of

[1] School of Information Science and Engineering, Fujian University of Technology, Fuzhou, 350118, China.

[2] School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha, 410004, China.

[3] School of Civil Engineering, Changsha University of Science & Technology, Changsha, 410000, China.

[4] Department Biomedical Engineering, University of Reading, Reading, RG6 6AY, UK.

[5] Computer Science Department, Community College, King Saud University, Riyadh, 11437, Saudi Arabia.

[6] Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Shebin-El-Kom, 32511, Egypt.

[*] Corresponding Author: Lei Wang. Email: Leiwang@csust.edu.

Received: 16 May 2020; Accepted: 16 July 2020.

the information technology industry, after computers and the Internet. Its ubiquitous network characteristics make the interconnection of everything possible [Ren, Zhu, Sharma et al. (2019)]. Underpinning the acceleration and rapid maturity of sensor networks are the dam secure system [Mao, Zhang, Qi et al. (2019); Liu, Huang and Zhang (2018)], the Internet of Vehicles (IoV) [Cao, Jiang, Wang et al. (2020); Cao, Liu, Ma et al. (2019); Cao, Zheng, Ji et al. (2019)], the medical system [Zhang, Duan, Sun et al. (2019)], artificial intelligence [Rostami, Sangaiah, Wang et al. (2019)], intelligent transportation [Sun, Zhang, Zhang et al. (2019)], etc. There is no doubt that the era of sensor networks is rapidly approaching. Therefore, the application of sensor network technology is becoming increasingly demanding.

As one of the basic components of sensor networks, the network itself is characterized by a large amount of data collection and multiple transmission processes between nodes; thus, it often encounters security threats such as information leakage, information forgery, and unauthorized access [Wang, Gao, Wang et al. (2019); Xia, Ying, Lin et al. (2019)]. Because the network nodes' computing and storage capabilities are strictly limited, many mature data security protection technologies in traditional networks cannot be applied directly to sensor networks [Ge, Liu, Xia et al. (2019); Wang, Gao, Zhou et al. (2020); Okhovvat and Kangavari (2019)]. Moreover, privacy protection and the effectiveness of data sources play a key role in data mining. The question of how to ensure the legitimacy of information sources in the network, along with the security of information transmission, has become a vital problem that must be resolved as sensor networks become more popular.

To improve sensor network data security, some scholars have integrated blockchain technology into sensor networks [Ren, Zhu, Sharma et al. (2019); Ren, Liu, Ji et al. (2018)]. In blockchain technology, the data layer of the blockchain uses a Merkle tree to store data. However, Merkle trees are affected by a number of disadvantages. Firstly, they can only provide member proof, but cannot provide non-member proof. Moreover, Merkle tree-based data storage takes up a large amount of memory. By contrast, the cryptographic accumulator is versatile and compact, and can be used both to achieve non-member proof and reduce storage overhead.

This research paper studies the accumulator-based blockchain data storage mechanism, and further proposes a secure data sensor network storage method based on blockchain. Moreover, it proposes the concept of an unbounded accumulator, uses bilinear pairing to construct a specific scheme, and demonstrates its security.

The remainder of this paper is structured as follows. Section 2 presents some related work about sensor networks and their architecture, blockchain technology, and the bilinear mapping accumulator. In Section 3, the data security issues of sensor networks are described. Section 4 presents and describes the secure data storage of sensor networks based on blockchain. In Section 5, moreover, the comparison and performance analysis of the scheme are outlined. Finally, conclusions are provided in Section 6.

2 Related work

2.1 Sensor networks

Sensor networks are regarded as the third revolution in the growth of the information technology industry, after computers and the Internet. A sensor network is the Internet that connects things to other things [Fang, Li, Yun et al. (2019)]. Due to the rapid developments in information technology, the sensor network has become the forefront of technological development and has been used widely in various scenarios. Countless intelligent terminals in the sensor network transmit the massive amount of sensed data to the data center via complex information communication channels. The data center has huge storage and processing capabilities; this can be abstracted into a "terminal transmission pipeline cloud" architecture, which also corresponds to object. The three logical layers of the networking architecture are as follows: perception layer, network layer, and application layer [Zhou, Liu, Tang, et al. (2019); Zhao, Zhang, Wang et al. (2018); Medhane, Sangaiah, Hossain et al. (2020); Sangaiah, Darshan, Kumar et al. (2020)]. The architecture is as illustrated in Fig. 1 below.

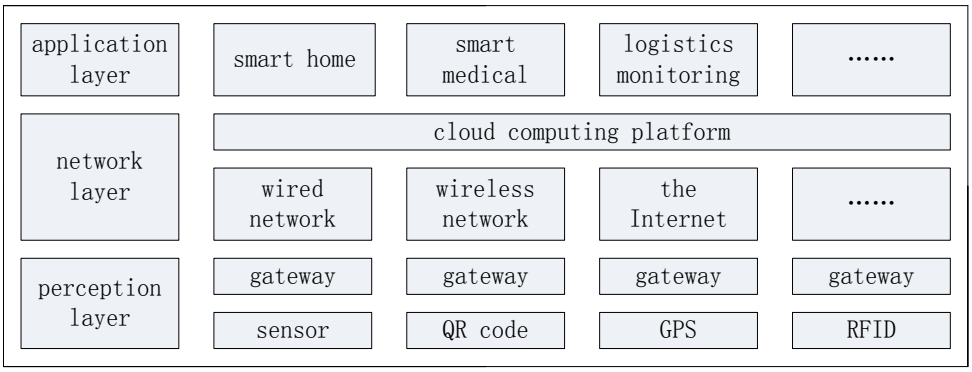

Figure 1: Sensor network architecture

The perception layer enables the sensor networks to identify things and collect information. The perception layer mainly includes sensors, RFID, GPS, and QR codes. After the relevant data is collected by the perceptual layer, it will be transmitted to the application layer through the network layer; the relevant nodes will then be controlled accordingly after the instructions are received.

The network layer is the part that connects the perception layer and the application layer. The data collected by the perception layer is transferred to the application layer through the network layer, while the instructions issued by the application layer are also transmitted from the network layer to the perception layer. The network layer principally uses wireless network technology, mobile communication technology, and wired network technology. Its core elements are network convergence technology and long-distance communication technology.

The application layer is the top layer and the most fundamental part of the sensor network architecture. This layer receives data from the perception layer through the network layer,

and further analyzes and processes the data. The information processing conducted within the application layer works mainly to issue operation instructions to the perception layer.

2.2 Blockchain technology

Blockchain technology is a new form of distributed ledger technology capable of realizing trusted transactions of the intermediary within an environment of mutual distrust [Ren, Leng, Cheng et al. (2019)]. When compared with traditional database technology, the blockchain technology boasts anti-forgery and anti-tampering features, and is consequently praised as a new technology that that can bring social change. In December 2016, the Chinese government first included blockchain technology in the "Thirteenth Five-Year Plan" National Informatization Plan [Mermer, Zeydan and Arslan (2018)], which aims to strengthen the fundamental research and development of new technologies and the layout of all employees. The blockchain can be interpreted as a type of decentralized distributed database that does not depend on any institution or administrator. The role of the blockchain is to store information. The data contained in the database is maintained by the nodes of the entire system. Anybody can access the blockchain network and become a data node. If data is written to a data node, this node will broadcast the written data information to neighboring nodes; these will then broadcast the information to their own neighboring nodes, until the information is eventually broadcast to all nodes in the entire network. Finally, all nodes will synchronize the information to guarantee its consistency.

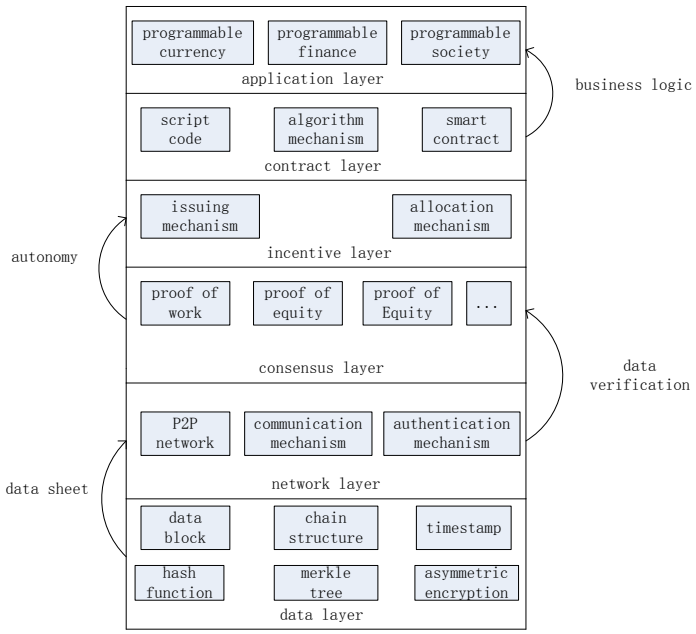

Figure 2: Blockchain infrastructure model

The basic model of the blockchain technology architecture is presented in Fig. 2. The blockchain system consists of a network layer, a data layer, a consensus layer, an incentive layer, a contract layer, and an application layer [Zhang, Zhong, Wang et al.

(2020)]. More specifically, the data layer encapsulates the underlying data and basic algorithms such as data encryption and timestamps, while the network layer comprises distributed networking, data transmission, and data verification mechanisms. The consensus layer mainly encapsulates various network node consensus algorithms. The incentive layer incorporates economic factors into the blockchain technology system; these mainly includes the economic incentive issuing and distribution mechanisms. The contract layer encapsulates algorithms, scripts and smart contracts [Singla, Malav, Kaur et al. (2019)], which form the basis of the blockchain programmable features. The application layer encapsulates numerous application scenarios and blockchain cases. In this model, the timestamp-based blockchain structure, the consensus mechanism of distributed nodes [Swan (2015)], the economic incentive based on consensus computing power, and flexible and programmable smart contracts are the greatest representative improvements of blockchain technology.

The Bitcoin blockchain is a chained data structure made up of sequential blocks. Each block consists of a block header (Header) and a block body (Body). The block header encapsulates the current bitcoin protocol version number, the hash value of the previous block (this hash value is calculated with reference to the size of the previous block, rather than all previous blocks), as well as the nonce, timestamp, and root hash value required by the current block's PoW (Proof of Work) [Ren, Leng, Cheng et al. (2019); Ren, Leng, Zhu et al. (2019)]. The bitcoin network can dynamically adjust the difficulty of the PoW process. The miner who first finds the correct random number nonce and has verified it by all miners will be given the right to record the current block. The body of the block contains both the number of transactions in the current block and all verified transaction records generated during the block creation process.

2.3 Accumulator based on bilinear pairing

Let G_1,G_2 be two cyclic multiplication groups of primes of order p generated by g_1 and g_2, and there is isomorphismψ: $G_2 \to G_1$ such that $\psi(g_2) = g_1$. Here, G_m is a cyclic multiplication group with the same order p, while e: $G_1 \times G_2 \to G_m$ is a bilinear pair with the following characteristics:

Bilinearity: $e(P^a, Q^b) = e(P, Q)^{ab}$ for all $P \in G_1, Q \in G_2, a, b \in Z_p$.

Non-degeneracy: $e(g_1, g_2) \neq 1$.

Computability: For all $P \in G_1, Q \in G_2$ has an effective algorithm to calculate (P, Q).

In this setup, $G_1 = G_2 = G$ and $g_1 = g_2 = g$. The bilinear pairing instance generator is a probabilistic polynomial time algorithm that takes the secure parameter 1^k as input and outputs a uniform random tuple of bilinear pairing parameters$t = (p, G, G_M, e, g)$. This includes a prime number p of size k, a cyclic addition group G_1 of order p, a multiplication group G_M of order p, and a bilinear map e: $G_1 \times G_2 \to G_M$ and a generator P of G_1.

Bilinear Mapping Accumulator [Ren, Qi, Liu et al. (2020)]: A bilinear mapping accumulator provides a short computational proof of membership or non-membership for elements that do or do not belong to a set. The bilinear mapping accumulator works as

outlined below. Given a set of n elements: $X = \{x_1, x_2 \ldots x_n\}$, the accumulator value is determined as follows:

$$\text{acc(X)} = g^{(x_1+s)(x_2+s)\ldots(x_n+s)}. \quad (1)$$

Here, g is the generator of the G group with prime order p, while $s \in Z_p^*$ is a randomly selected value that constitutes the trapdoor in the scheme. According to the accumulated value $acc(X)$, each element x in X has a member ID person W_x, such that:

$$W_X = g^{\prod_{x_j \in X: x_j \neq x}(x_j+s)}. \quad (2)$$

Therefore, given the accumulated value acc(X) and the witness W_x, we verify that x is a member of X by checking that $\text{e}(W_x, g^s g^x)$ is equal to $e(acc(X), g)$.

q-strong Diffie-Hellman hypothesis: Given a tuple of uniformly randomly generated bilinear pairing parameters $t = (p, G, G_M, e, g)$ and the elements $G_M, g, g^s, g^{s^2}, \ldots, g^{s^q}$ for some s that is randomly selected from Z_p^* for computationally bounded opponents A to $c \in Z_p$ and the output $(c, g^{1/(s+c)})$. This is a very difficult computation; that is, the probability can be neg (k), and it can be ignored in the secure parameter k.

3 Data security issues of the sensor networks

The security and privacy of the sensor networks are the key focus areas of sensor network security research. However, because these sensor networks lack a mutual trust mechanism between devices, all devices are required to check with the data of the sensor networks center; once the database collapses, this will result in significant damage to the entire sensor network. Moreover, there are a large number of data collection and transmission processes between sensor nodes in the sensor networks. Therefore, such systems often encounter security threats, such as information leakage, information forgery and unauthorized access. The structure of the sensor network usually consists of three parts: namely, the perception layer, the network layer, and the application layer. The following sections analyze the data security issues associated with each layer.

3.1 Perception layer security analysis

Sensor nodes are vulnerable to eavesdropping or control. The sensor nodes of the sensor network are characterized by simple functions, low processing power, and low energy consumption, and cannot autonomously implement comprehensive security protection. Moreover, the number of node groups is large, making management and control difficult and causing the system to be prone to omissions, which gives attackers an opportunity. Therefore, the communication information of the node is easy to eavesdrop on; moreover, it may even be possible to control the nodes so that the wrong information is sent, causing confusion in the network information [Wang, Shao, Gao et al. (2019); Zhou and Luo (2018)]. In addition, if the gateway node is eavesdropped on or controlled, this will lead directly to network paralysis and the leaking of all information in the network [Wang, Wang, Zheng et al. (2018)].

Node camouflage: Due to the vulnerability of the nodes and the variability of the network topology, an attacker could analyze a node to obtain its identity and password information [Wang, Gu, Liu et al. (2019); Gong, Yang, Xue et al. (2018)], tamper with

the software and hardware, and then capture the node to disguise themselves as a legitimate user. Malicious attacks have resulted in errors in the node data collection process [Wang, Wu, Liao et al. (2020); Wang, Yang, Wang et al. (2020)]; these include the monitoring of user information, replacing devices, publishing false information, and launching DoS attacks.

3.2 Network layer security analysis

Due to the small amount of data transmitted by devices in the sensor networks, complex encryption algorithms are generally not used to protect the data. This results in data being stolen, tampered, attacked, and illegally accessed by the network during transmission, as well as precluding data from being shared in secret [Zhang, Shi, Hu et al. (2018)]. Multiple denial-of-service attacks, man-in-the-middle attacks, virus intrusions, and attacks using sniffing tools and system vulnerabilities can also result.

3.3 Application layer security analysis

The sensor network application layer stores a large amount of user data. Accordingly, it is necessary to consider several security issues in the application layer: for example, how to effectively store data in order to avoid data loss or damage, how to isolate data for multi-tenant applications, how to avoid high latency and high energy consumption caused by large amounts of data [Li, Yan, Chen et al. (2019)], and how to quickly recover data after failures.

Data access rights and user authentication: The application layer is the layer that interworks directly with users and provides users with data access rights. Therefore, key points of security for sensor network applications include a sound authentication mechanism, access permission settings, authorization management protocol, data integrity analysis [Wan, Chen and Zhang (2019)], and the isolation of intrusion by illegal users.

User privacy leakage issues: Privacy issues represent the biggest obstacle to the implementation of sensor networks. The sensor network is involved with many aspects of users' lives. Once information is leaked, users' property, information security, and personal privacy can easily be violated [Xu, Zhang, Fu et al. (2019)]. Ensuring information privacy is therefore the first problem that must be solved if the development of sensor networks is to be promoted.

4 Secure data storage of sensor networks based on blockchain

4.1 Data storage of sensor networks based on blockchain

Firstly, this paper constructs a secure data storage mechanism for sensor networks based on blockchain, which is illustrated in Fig. 3 below.

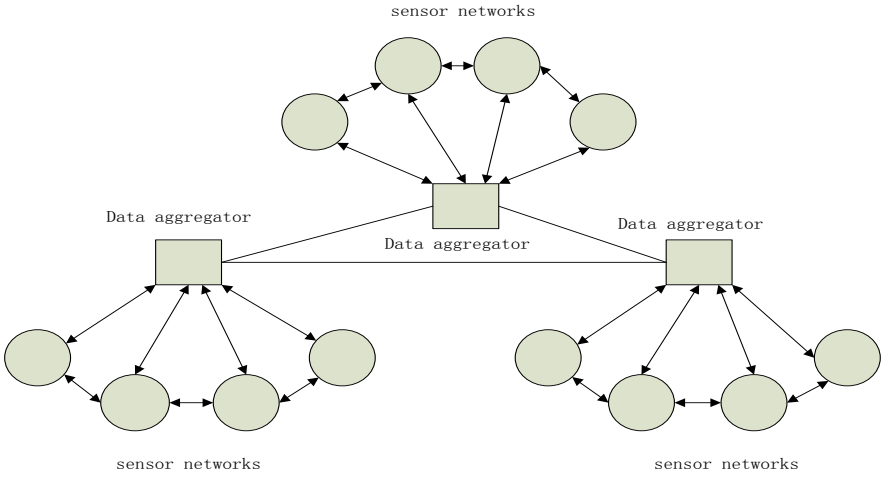

Figure 3: Secure data storage of sensor networks based on blockchain

As can be seen in Fig. 3, the blockchain-based sensor network data storage mechanism is mainly composed of two parts: sensor nodes and data aggregators. The sensor nodes collect data, while the data aggregators analyze and store the data, while each data aggregator is connected to form a $P2P$ network. The working principle is as follows: First, the sensor nodes in each area form a sensor network. Each sensor node collects device data and sends it to the data aggregator. In the next step, each data aggregator analyzes and verifies the sent data. Finally, after data analysis and judgment is complete, the data is stored securely through the blockchain.

4.2 Data storage mechanism based on accumulator in blockchain

Data in traditional blockchains is stored using Merkle hash trees, meaning that non-member proof cannot be provided. By contrast, the accumulator can provide non-member proof. Accordingly, this paper proposes to use an accumulator instead of the original Merkel tree in the block to build a data storage mechanism of blockchain based on the accumulator.

Each block contains a block header and a block body. In addition to replacing the Merkle tree in the block with an accumulator, this paper also allocates an accumulator to each node in the entire network, so that each node is a full node with two distinct functions. The accumulated value of the data is stored in the block header, which is uploaded by the data provider and then accumulated by the accumulator, after which the accumulated value is added. This accumulated value is the accumulated value of the accumulator on the node. The Merkle tree in the block thus becomes an accumulator. The data points $n_1, n_2, \ldots, n_m$ represent the data collected by the nodes over a period of time. At this time, the hash value is not computed in the block body, but rather in the accumulated value. The accumulated value obtained is stored in the block header. The advantage of this approach is that it can reduce the amount of storage memory that was previously used to store hash values for each layer of the Merkle tree, and can now store an accumulator that can also provide non-member proof.

The value accumulated by the accumulator on each node represents the value of all data in the entire blockchain; that is, assuming that the block is the n^{th} block and the accumulated value is A, a new block is added at this time. For $n + 1$ blocks of data, the accumulated value A is also changed accordingly. The accumulated value of each block is the accumulated value of the data collected by this block in a certain time.

The blockchain is jointly maintained by many network nodes. The improved blockchain proposed in this paper comprises an accumulator for each block, while each node also has an accumulator. Both accumulated values are stored in the block header, and each block connects with every other block by finding the hash value of the block header.

First, the block data is created: to achieve this, a certain area in the network generates a set of data, along with a signature unique to the place where the data is published; subsequently, the data is broadcast to the entire network, and other places do the same. That is, the data is generated continuously. At this time, each of the network nodes collects the data broadcast over a period of time, verifies the legitimacy of the data source, passes the verification, and arranges the data in a certain order.

In the next step, a new block is created. After each network node has collected the data, a new block is created. At this time, the accumulator in the block generates the accumulated value in the block, after which the accumulated value of the accumulator data on the node changes accordingly. Both of these accumulated values are stored in the block header, after which the block header information is combined into a string. A 250 binary number is obtained through the hash function twice, and the result is then generated. The difficulty value setting is met if the first few digits are 0; if it is not satisfied, it must recompute by adjusting the nonce value until it the setting is met. This is as follows:

$$H(block\ header) \leq target. \tag{3}$$

Once a node in the network determines that the new block is successfully created, the node will broadcast a block success message to the entire network, which will be received by the other nodes.

The second element is node verification. When a node receives a message from a new block, the node will verify it. At this time, the content of the verification comprises two parts: whether the data in the block is included in the accumulated value of the node, and whether the hash value of the block head is less than the target difficulty. Once each node passes the verification, a consensus is reached, and other network nodes agree with this newly generated block. At this time, the node verification process also includes the step that, after the acquisition of the intelligence demander is completed, the information will be broadcast to all network nodes through the *P2P* network. The nodes throughout the entire network then agree on the information request. After the request is completed, an intelligent transaction log will be generated. After the authentication of the blockchain sharing platform has been obtained, it becomes a historical data point that can be queried.

Finally, after the node verification is complete, the hash value obtained by the block header is connected to the hash value of the previous block header, after which the new block is successfully added to the blockchain.

In this paper, accumulators are used in the block rather than the Merkle tree, and

accumulators is added to the nodes; this can not only reduce the storage memory significantly, but is also convenient for both member and non-member verification. For example, the bilinear accumulators can be used to prove membership. Directly through the accumulator can quickly provide member proof, as shown in Eq. (2). It is not necessary to apply for a node, as with a Merkle tree, and start computing the various hash values involved from the leaf node. If the node verification is light, it will be necessary to apply for other hash values from other nodes.

4.3 Unbounded accumulator definition and security model

Because the accumulator has an upper limit on its accumulation elements, it is difficult to meet the requirements for large-scale storage. Accordingly, we expand the capacity of the blockchain. This paper proposes an unbounded accumulator, meaning that an unlimited number of elements can be accumulated by the accumulator.

4.3.1 Unbounded accumulator definition

The accumulator scheme has the following attribute definitions. We use Z_N to represent the domain for accumulating values and Z_M to represent the domain of the accumulator.

Effective generation: There is an effective probabilistic algorithm that defines the function *f*: $Z_M \times Z_N \to Z_M$ when inputting the security parameter k to generate the accumulator-specific key pair (sk, pk), where sk is the trapdoor for f.

Effective evaluation: There is an effective algorithm to calculate $f(acc, x)$.

Quasi-commutativity: $f(f(acc, x_1), x_2) = f(f(acc, x_2), x_1), \forall_{x_1,x_2} \in Z_N, acc \in Z_M$.

Supposing that it is not feasible to compute f without knowing sk, quasi-commutativity directly leads to a method that can be used to define witnesses. For example, $f(acc, x_1)$ can be used as a witness for x_2 accumulation. Nonetheless, as outlined below, it makes more sense to provide a more abstract definition of the algorithm for the accumulator, as there are constructs that are not suitable for this representation.

Definition 1 (Unbounded Accumulator): Unbounded accumulators are static, dynamic, and general-purpose accumulators that have the following attributes. For the static, dynamic, and universal accumulators, let the parameter $t = \infty$ in the key generation algorithm; that is, the number of accumulated elements is infinite.

4.3.2 Security model

In this section, this paper introduces a security model for the unbounded accumulator. A successful secure unbounded accumulator scheme must be correct and collision-free. The probability of an attack occurring, that is, the probability of success is very small, and can be negligible. The unbounded accumulator is considered secure. Notably, the Ver algorithm will always return true for all honestly generated keys, all accumulated values and all witnesses computed honestly. We emphasize the need to maintain correctness even when executing all algorithms without using sk. Here, 'correctness' signifies that the unbounded accumulator is able to output the correct accumulated value. Moreover, 'collision-free' can be informally defined as a state such that finding non-cumulative

value for witnesses is unfeasible, as is accumulative value for non-member witnesses. More formally:

Definition 2 (Unbounded accumulator without collision): if all PPT (probabilistic polynomial-time) opponent A function neg(.) are negligible, then $t \in$ {static Unbounded, dynamic Unbounded} and $u \in$ {general Unbounded, non-universal unbounded} password accumulators are collision-free:

$$\boldsymbol{pr}[(\boldsymbol{sk_{acc}}, \boldsymbol{pk_{acc}}) \leftarrow \boldsymbol{Gen}(\boldsymbol{1^k}, \boldsymbol{t}), \boldsymbol{\vartheta} \leftarrow \{\boldsymbol{\vartheta^t}, \boldsymbol{\vartheta^u}\}, (\boldsymbol{wit}^*_{x_i}, \underline{\boldsymbol{W}}^*_{x_i}, \boldsymbol{x}^*_i, \boldsymbol{X}^*, \boldsymbol{r}^*) \leftarrow \boldsymbol{A^{\vartheta}}(\boldsymbol{pk_{acc}}): \left(\mathbf{Ver}(\boldsymbol{pk_{acc}}, \boldsymbol{acc}^*, \boldsymbol{Wit}^*_{x_i}, \boldsymbol{x}^*_i, \mathbf{0})\right) = \boldsymbol{true} \wedge \boldsymbol{x}^*_i \notin \boldsymbol{X}^* \vee \left(\mathbf{Ver}(\boldsymbol{pk_{acc}}, \boldsymbol{acc}^*, \underline{\boldsymbol{W}}^*_{x_i}, \boldsymbol{x}^*_i, \mathbf{1})\right) = \boldsymbol{true} \wedge \boldsymbol{x}^*_i \in \boldsymbol{X}^*] \leq \boldsymbol{neg}(\boldsymbol{k})$$

where $acc^* \leftarrow Eval_{r^*}((sk, pk), X^*)$ and A have predictive access to ϑ^t and ϑ^u, and are defined as follows:

$$\boldsymbol{\vartheta^t} := \{\boldsymbol{\vartheta^{E(\cdot,\cdot,\cdot)}}\} \textbf{ if } \mathbf{t} = \textbf{Static unbounded}$$

$$\boldsymbol{\vartheta^t} := \{\boldsymbol{\vartheta^{E(\cdot,\cdot,\cdot)}}, \boldsymbol{\vartheta^{A(\cdot,\cdot,\cdot)}}, \boldsymbol{\vartheta^{D(\cdot,\cdot,\cdot)}}\} \ \mathbf{t} = \textbf{dynamic unbounded}$$

$$\boldsymbol{\vartheta^u} := \{\boldsymbol{\vartheta^{W(\cdot,\cdot,\cdot)}}, \boldsymbol{\vartheta^{\underline{W}(\cdot,\cdot,\cdot)}}\} \text{ if u = Universal unbounded}$$

$$\vartheta^u := \{\vartheta^{W(\cdot,\cdot,\cdot)}\} \text{ u= Non-universal unbounded}$$

Therefore, ϑ^E, ϑ^A, and ϑ^D represent the predictions of Eval, Add, and Delete, respectively, which allows opponents to query them any number of times. In the case of random unbounded accumulators, the opponent outputs randomness r^*, while the deterministic unbounded accumulators omit r^*. Similarly, the opponent can control the randomness of ϑ^E for a random unbounded accumulator. Therefore, ϑ^E uses an additional parameter for r (the prophets ϑ^w and $\vartheta^{\underline{w}}$ allow the opponent to obtain member witnesses and non-member witnesses, respectively. Therefore, the environment will track all prediction queries; if the call to it is inconsistent with the previous query, the corresponding prediction $\perp$ is returned. Furthermore, we can assume that the opponent outputs either a member witness $Wit^*_{x_i}$ or a non-member witness $\underline{W}^*_{x_i}$. If the unbounded accumulator is non-universal, you only need to ignore the non-membership related parts.

Definition 8 (secure unbounded accumulator): If the unbounded accumulator is correct and collision-free, it is secure.

4.4 Unbounded accumulator scheme based on bilinear pairs

The unbounded accumulator proposed in this paper is implemented by means of bilinear pairs. The scheme consists of seven parts: $\{KeyGen, Eval, Wit, Ver, Add, Delete, UpdWit\}$. This scheme is able to add and delete elements, and there is no limit placed on the number of elements accumulated by the accumulator. The specific scheme is described in section 4.4.1.

4.4.1 Specific scheme

$KeyGen$: A probabilistic algorithm used to generate initial parameters. First run a bilinear pairing instance generator on input k to generate a uniform random tuple of bilinear pairing parameters $t = (p, G, \mathcal{G}, e, g)$; then randomly select a number $s \in Z_p^*$ as a trapdoor; then select N to denote the total number of elements to be accumulated, where $N = \infty$ is the upper limit of the accumulated elements; finally, return the key pair $h = \{sk, pk\}$.

$Eval$: A probabilistic algorithm for accumulation value. Given a set of elements $X = \{x_1, x_2, \dots, x_n\}$ and h as input, compute $acc(X) = g^{(x_1+s)(x_2+s)\dots(x_n+s)}$. Find the accumulated value. Finally, output the accumulated value $acc(X)$ and auxiliary information a_c.

Wit: Probability algorithm for generating witnesses that correspond to each element. Given a set of elements $X = \{x_1, x_2, \dots, x_n\}$, with auxiliary information a_c and h as input, calculate $W_{x_i} = g^{\prod_{x_j \in X: x_j \neq x_i}(x_j+s)}$. The witnesses of each element are output as (W_{x_i}, x_i).

Ver: A deterministic algorithm for verifying the identity of members through witnesses. Given an element x, its witness W, and the accumulative value $acc(X)$ and h, check whether the element x belongs to the set X represented by accumulative value $acc(X)$ and $W^{(x+s)} = acc(X)$; if yes, return Yes, otherwise return No.

Add: Element addition algorithm. Given an additional element x^+, auxiliary information a_c, accumulative value $acc(X)$ and h as inputs, calculate $acc(X') = acc(X)^{(x^++s)}$, $W_{x_i}^{x^+} = g^{\prod_{x_j \in X: x_j \neq x_i}(x_j+s)(x_i+s)}$. Then output a new accumulated value $acc(X')$, witness $(W_{x_i}^{x^+}, x^+)$ and auxiliary information a_c and a_u.

$Delete$: Element deletion algorithm. Once the elements x^- to be deleted in the given set, accumulative value $acc(X)$, auxiliary information a_c and h are input, compute $acc(X') = acc(X)^{\setminus(x^-+s)}$. A new accumulated value $acc(X')$ is then output, which corresponds to the set $\{X \setminus x^-\}$ and auxiliary information a_c and a_u.

$UpdWit$: Witness update algorithm. Given a witness W_{x_i}, auxiliary information a_c and h as input, add $W'_{x_i} = g^{\prod_{x_j \in X \cup x^+: x_j \neq x_i}(x_j+s)}$; it then outputs a new witness W'_{x_i} for each element x_i. When deleting $W'_{x_i} = g^{\prod_{x_j \in X \setminus x^-: x_j \neq x_i}(x_j+s)}$, it outputs a new witness W'_{x_i} for each element x_i.

4.4.2 Security analysis

Correctness:

The 'correctness' property of the accumulator scheme is simply to indicate that if the element x belongs to the accumulation set X, and if the corresponding witness W has been calculated using Wit and $UpdWit$, the verification process should pass. The scheme $\{KeyGen, Eval, Wit, Ver, Add, Delete, UpdWit\}$ is correct. Moreover, if there is a sufficiently large $k \in N$, for all $\{sk, pk\}$ output by the algorithm $KeyGen$, and for all accumulated values, the witness output via $Eval$, Wit, $UpdWit$ without upper limit accumulator is correct. When the elements of the collection change, most of them will change accordingly.

Proof We refer to a result as 'correct' if the correctly accumulated value has verifiable witnesses. When it is a non-member, the non-membership proof computed by *Wit* is (A_x, B_x, x) eligible. Because

$e(acc(X), A_x)e(g^s g^x, B_x) = e\left(g^{\prod(x_i+s)}, A_x\right)e(g^s g^x, B_x) = e(g, g)$.

Moreover, $A_x = g^{\alpha(s)}, B_x = g^{\beta(s)}$ makes

$$(\prod_{i=1}^{n} x_i + s)\alpha(s) + (x + s)\beta(s) = 1 \quad (4)$$

Second, it shows that the Ver algorithm is correct for the accumulator. Let set $X = \{x_1, x_2, \dots, x_n\}$, while $acc(X)$ is the corresponding accumulated value. Here, x is considered to be the i -th element of the set, because $acc(X) = g^{(x_1+s)(x_2+s)\dots(x_n+s)}$, $W_{x_i} = g^{\prod_{x_j \in X: x_j \neq x_i}(x_j+s)}$. Due to the following formula,

$$W^{(x+s)} = W_{x_i}^{(x_i+s)} = g^{\prod_{x_j \in X: x_j \neq x_i}(x_j+s)(x_i+s)} = g^{(x_1+s)(x_2+s)\dots(x_n+s)} = acc(X) \quad (5)$$

So $W^{(x+s)} = acc(X)$. In the accumulator, if the element x is accumulated to the accumulated value, the witness can provide a valid proof for X.

$$W_i'^{x_i} = W_{x_i}^{x_i x^+} = g^{\prod_{x_j \in X: x_j \neq x_i}(x_j+s)(x_i+s)(x^+ +s)} = acc(X)^{(x^+ +s)} = acc(X') \quad (6)$$

Moreover, when deleting elements, the correctness can be verified in the same way.

Security:

Security is present if, for all sufficiently large $k \in N$, for all $\{sk, pk\}$ output by the algorithm $KeyGen$, and for all adversaries constrained by polynomials, the probability is negligible and the unbounded accumulator is secure. The bilinear q-strong Diffie-Hellman also assumes that the scheme is secure.

Bilinear to the main security requirements of the unbounded accumulator: anti-collision. Let k be the secure parameter; moreover, let $t = (p, G, \mathcal{G}, e, g)$ be the uniformly randomly generated bilinear pair parameter tuple, where p is the k-bit prime number. Given the element $g, g^s, g^{s^2}, \dots, g^{s^q} \in G$ for some s randomly selected from Z_p^* and any k-bit element $X(q \geq |X|)$. The security is explained by the sum value and the witness; that is, the sum value and the witness that an opponent can find out from another set is equal to the original value, but the probability can be ignored in the security parameter k.

Theorem 1. Given a set of elements X, select some s from Z_p^* and t; considering only the case of $g, g^s, g^{s^2}, \dots, g^{s^q} \in G$, A can find a set $X' \neq X$ such that $acc(X') = acc(X)$. However, this probability is negligible in the secure factor k.

Proof 1 Suppose that A finds such a set X'; this means that A finds another set $X' = \{x_1', x_2', \dots, x_{n'}'\} \neq X = \{x_1, x_2, \dots, x_n\}$; thus,

$$g^{(x_1+s)(x_2+s)\dots(x_n+s)} = g^{(x_1'+s)(x_2'+s)\dots(x_{n'}'+s)} \quad (7)$$

Namely,

$$acc(X)' = acc(X) \quad (8)$$

which implies that:

$$A^{(x_j'+s)} = g^{(x_1+s)(x_2+s)\dots(x_n+s)} \quad (9)$$

Where:

$$A = g^{\prod_{i \neq j}(x'+s)} \quad (10)$$

For some x_j' that does not form part of the original set, there is the following formula:

$$\prod_n = (x_1 + s)(x_2 + s) \dots (x_n + s) \quad (11)$$

Since $x_j' \notin X$, we have $(x_j' + s)$ not dividing $\prod n$. Thus, A can find c and P such that:

$$\prod_n = c + P(x_j^{'} + s) \tag{12}$$

The following formula can therefore be obtained:

$$g^{1/(x+s)} = [A[g^p]^{-1}]^{c^{-1}} \tag{13}$$

Theorem 2. Regarding the security of the bilinear mapping accumulator, let k be the security parameter. Moreover, let $t = (p, G, \mathcal{G}, e, g)$ be a uniformly and randomly generated bilinear pair parameter tuple, where p is a *k*-bit prime number. Given the elements $g, g^s, g^{s^2}, \ldots, g^{s^q} \in G$ for some s that is randomly selected from Z_p^* and any *k*-bit element $X(q \geq |X|)$, assume a polynomial time algorithm for any of the following tasks:

The algorithm outputs $x \notin X$ and W such that $e(W, g^s g^x) = e(acc(X), g)$.

The algorithm outputs $x \in X$, A, and B such that $e(acc(X), A)e(g^s g^x, B) = e(g, g)$.

Thus, there is a polynomial-time algorithm for breaking the bilinear q-strong Diffie-Hellman assumption.

Proof 2 Let $X = \{x_1, x_2, \ldots, x_n\}$ and let $x \notin X$. Suppose an algorithm finds W such that $e(W, g^s g^x) = e(acc(X), g)$. This implies that:

$$e(W, g)^{s+x} = e(g, g)^{(x_1+s)(x_2+s)\ldots(x_n+s)} \tag{14}$$

Now note that the formula is shown in Eq. (11).

Since x ∉ X and $(s + x)$ does not divide $\prod n$, the values of c and P can be calculated. Accordingly, the following formula is obtained: $\prod_n = c + P(x + s)$. Thus, this algorithm can output $(x, e(g, g)^{\frac{1}{s+x}})$ as $(x, \left[e(W, g)e(g, g)^{-P}\right]^{c^{-1}})$. For non-member proof, it outputs $e(g, g)^{1/(s+x)}$ as $e(W_x, A)e(g, B)$. Since $x \in X$, $e(acc(X), A)e(g^s g^x, B) = e(g, g)$ is obtained, where W_x is a membership witness. Therefore, the q-strong Diffie-Hellman assumption can be broken in both cases.

5 Comparison and performance of the scheme

In this section, the scheme is compared and executed, which will illustrate the advantages of the proposed scheme.

5.1 Scheme comparison

In this section, a comparison of previous works and the scheme presented in this paper is conducted. as shown in Tab. 1 below:

Table 1: Comparison of schemes

Scheme	D	U	Un	Mem	Non-mem	A	De	AA	Mem*	Non − mem*
DT [Damgård and Triandopoulos (2008)]	√	√	—	√	√	—	—	N	$\vartheta(1)$	$\vartheta(1)$
BC [Boneh and Gibbs (2014)]	—	—	—	√	—	—	—	N	$\vartheta(logt)$	$\vartheta(logt)$

CHKO [Camacho, Hevia, Kiwi et al. (2008)]	—	√	—	√	—		—	—	N	$\vartheta(logt)$	$\vartheta(logt)$
CF [Catalano and Fiore (2013)]	√	√	—	√	√		√	√	N	$\vartheta(1)$	$\vartheta(1)$
CKS [Camenisc, Kohlweiss, and Soriente (2009)]	√	—	—	√	—		—	√	N	$\vartheta(1)$	$\vartheta(1)$
AWSM [Au, Wu and Susilo (2007)]	√	—	—	√	—		√	—	N	$\vartheta(1)$	—
LLX [Li, Li and Xue (2007)]	√	√	—	√	√		√	—	N	$\vartheta(1)$	$\vartheta(1)$
ATSM [Au, Tsang, Susilo et al. (2009)]	√	√	—	√	√		—	—	N	$\vartheta(1)$	$\vartheta(1)$
This paper	√	√	√	√	√		√	√	∞	$\vartheta(1)$	$\vartheta(1)$

Legend: D=dynamic, U=universal, Un=unbounded, Mem=membership, Non-mem=non-membership, A=add, De=delete, AA=accumulator that accumulates the number of elements, Mem^*=membership witness size, $\mathrm{Non-mem}^*$=non-membership witness size; √=yes, —=not available.

From the analysis in Tab. 1 above, it can be seen that the accumulated elements of the previous scheme have some limitations. The schemes proposed by Boneh et al. [Boneh and Gibbs (2014); Camacho, Hevia, Kiwi et al. (2008)] cannot provide the addition, deletion and non-member proof features, while the scheme devised by Au et al. [Au, Wu and Susilo (2007)] can provide addition, but not deletion and non-member verification. Moreover, while the scheme proposed by Damgård et al. [Damgård and Triandopoulos (2008); Man, Patrick, Tsang et al. (2009)] can provide non-member authentication, it does not provide addition or deletion features. Camenisc et al.'s [Camenisc, Kohlweiss and Soriente (2009)] scheme can provide deletion, but no addition or non-member verification features. In addition, Li et al. [Li, Li and Xue (2007)] provide addition and non-member verification features in their scheme, but not deletion features, while Catalano et al. [Catalano and Fiore (2013)] can provide addition, deletion and non-member verification features. Notably, not only does the scheme proposed in this paper feature addition, deletion and non-member verification, there is also no limit to the number of accumulations.

5.2 Performance analysis

In this section, the performance of the unbound accumulator scheme is preliminarily studied. The results are illustrated in Figs. 4 and 5 below.

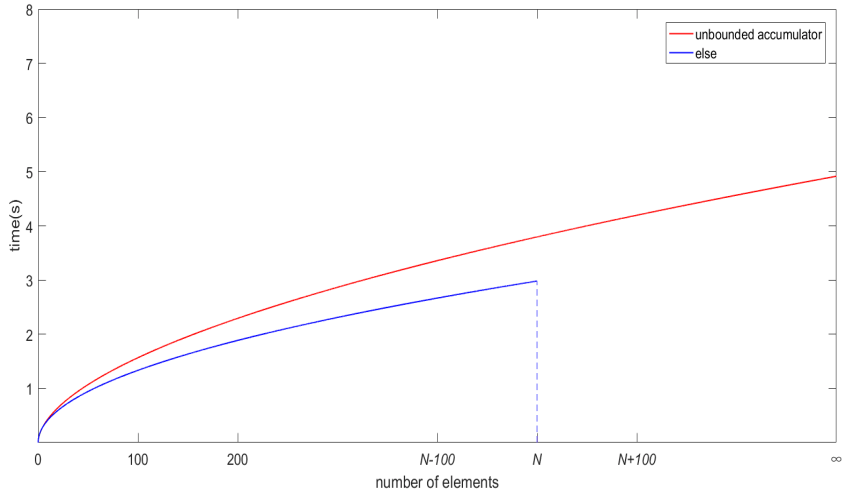

Figure 4: Number of elements in accumulator

In Fig. 4, the *X* axis represents the accumulator elements, while the *Y* axis represents the elapsed time. Moreover, the red line represents an unbounded accumulator; that is, while there is no limit to the number of elements accumulated, this comes at the expense of time. For its part, the blue line represents other types of accumulators, and the accumulated elements are limited.

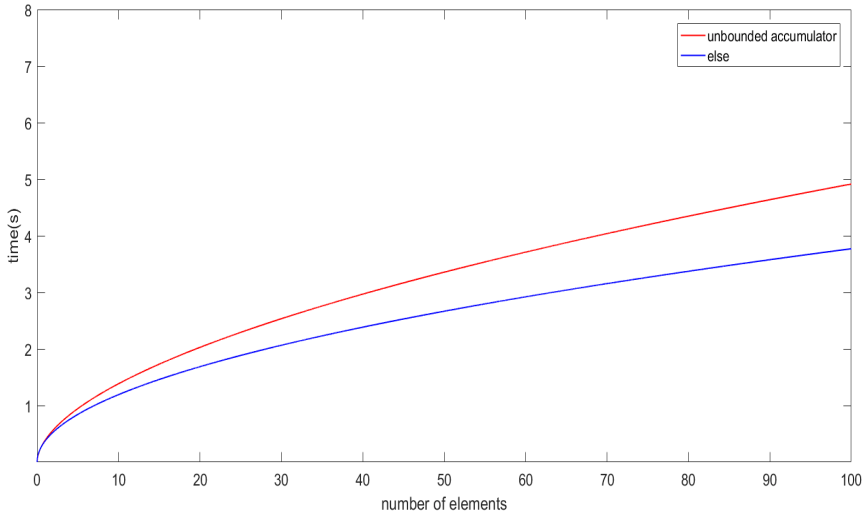

Figure 5: Element update time in accumulator

In Fig. 5, the *X* axis represents the accumulator accumulating elements, while the *Y* axis represents the time taken to update the witness. The red line represents an unbounded accumulator, while the blue line represents other types of accumulators. Here, there is no limit to the number of elements that the unbounded accumulator can accumulate. However, this comes at the cost of a longer time required to update the witness elements.

6 Conclusion

This paper first analyzes the challenges of sensor network data security, then analyzes the blockchain technology and proposes a blockchain-based sensor network data security storage mechanism. The Merkel tree utilized in the blockchain data layer cannot provide proof of non-membership; moreover, the accumulator has the advantages of sturdiness, universality, and compactness, and can further provide non-member proof, reduce data storage memory, and better protect privacy. Therefore, this paper constructs a blockchain-based accumulator for sensor networks. However, there is a limit to the number of elements that can be accumulated by a traditional accumulator. Accordingly, to solve the problem of data storage expansion in the blockchain, the concept of an unbounded accumulator is proposed: this can not only provide non-member proof, but can also resolve the limitation of the number of elements that can be accumulated by traditional accumulators. Finally, the effectiveness of the proposed scheme is proved through scheme comparison and performance analysis.

Funding Statement: This work is supported by the NSFC (61772454), the Researchers Supporting Project No. RSP-2020/102 King Saud University, Riyadh, Saudi Arabia. This work is also funded by National Key Research and Development Program of China (2019YFC1511000).

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

Au, M. H.; Tsang, P. P; Susilo, W.; Mu, Y. (2009): Dynamic universal accumulators for DDH groups and their application to attribute-based anonymous credential systems. In: Fischlin, M. (eds.), *Topics in Cryptology, Lecture Notes in Computer Science*, vol. 5473, pp. 295-308. Springer.

Au, M. H.; Wu, Q. H.; Susilo, W.; Mu, Y. (2007): Compact e-cash from bounded accumulator. In: Abe, M. (eds.), *Topics in CryptologyLecture Notes in Computer Science*, vol. 4377, pp. 178-195. Springer.

Boneh, D.; Gibbs, H. C. (2014): Bivariate polynomials modulo composites and their applications. In: Sarkar, P., Iwata, T. (eds.), *Advances in Cryptology-ASIACRYPT, Lecture Notes in Computer Science*, vol. 8873, pp. 42-62. Springer.

Cao, D.; Jiang, Y. C.; Wang, J.; Ji, B. F.; Alfarraj, O. et al. (2020): ARNS: adaptive relay-node selection method for message broadcasting in the internet of vehicles. *Sensors*, vol. 20, no. 5, pp. 1338.

Cao, D.; Liu, Y. H.; Ma, X. M.; Wang, J.; Ji, B. F. et al. (2019): A relay-node selection on curve road in vehicular networks. *IEEE Access*, vol. 7, no. 1, pp. 12714-12728.

Cao, D.; Zheng, B.; Ji, B. F.; Lei, Z. B.; Feng, C. F. (2020): A robust distance-based relay selection for message dissemination in vehicular network. *Wireless Networks*, vol. 26, no. 3, pp. 1755-1771.

Camacho, P.; Hevia, A.; Kiwi, M.; Opazo, R. (2008): Strong accumulators from collision-resistant hashing. In: Wu, T. C., Lei, C. L., Rijmen, V., Lee, D. T. (eds.), *Information Security, Lecture Notes in Computer Science*, vol. 5222, pp. 471-486. Springer.

Catalano, D.; Fiore, D. (2013): Vector commitments and their applications. In: Kurosawa, K., Hanaoka, G. (eds.), *Public-Key Cryptography, Lecture Notes in Computer Science*, vol. 7778, pp. 55-72. Springer.

Camenisch, J.; Kohlweiss, M.; Soriente, C. (2009): An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In: Jarecki, S., Tsudik, G. (eds.), *Public Key Cryptography, Lecture Notes in Computer Science*, vol. 5443, pp. 481-500. Springer.

Damgård, I.; Triandopoulos, N. (2008): Supporting nonmembership proofs with bilinear-map accumulators. *IACR Cryptology ePrint Archive*. https://eprint.icar.org/2008 /538.

Fang, L. M.; Li, Y.; Yun, X. Y.; Wen, Z. Y.; Ji, S. L. et al. (2019): THP: A novel authentication scheme to prevent multiple attacks in SDN-based IoT network. *IEEE Internet of Things Journal*.

Ge, C. P.; Liu, Z.; Xia, J.; Fang, L. M. (2019): Revocable identity-based broadcast proxy re-encryption for data sharing in clouds. *IEEE Transactions on Dependable and Secure Computing*.

Gong, L.; Yang, B.; Xue, T.; Chen, J.; Wang, W. (2018): Secure rational numbers equivalence test based on threshold cryptosystem with rational numbers. *Information Sciences*, vol. 466, pp. 44-54.

Liu, T.; Huang, G.; Zhang, P. (2018): A user authentication protocol combined with the trust model biometrics and ECC for wireless sensor networks. *Intelligent Automation and Soft Computing*, vol. 24, no. 3, pp. 519-529.

Li, J. T.; Li, N. H.; Xue, R. (2007): Universal accumulators with efficient nonmembership proofs. In: Katz J., Yung M. (eds.), *Applied Cryptography and Network Security, Lecture Notes in Computer Science*, vol. 4521, pp. 253-269. Springer.

Li, G. S.; Yan, J. H.; Chen, L.; Wu, J. H.; Lin, Q. Y. et al. (2019): Energy consumption optimization with a delay threshold in cloud-fog cooperation computing. *IEEE Access*, vol. 7, pp. 159688-159697.

Mao, Y.; Zhang, J.; Qi, H.; Wang, L. (2019): DNN-MVL: DNN-multi-view-learning-based recover block missing data in a dam safety monitoring system. *Sensors*, vol. 19, no. 13, pp. 2895.

Medhane, D. V.; Sangaiah, A. K.; Hossain, M. S.; Muhammad, G.; Wang, J. (2020): Blockchain enabled distributed security framework for next generation IoT: An edge-cloud and software defined network integrated approach. *IEEE Internet of Things Journal*.

Mermer, G. B.; Zeydan, E.; Arslan, S. S. (2018): An overview of blockchain technologies: Principles, opportunities and challenges. *Signal Processing and Communications Applications Conference*, pp. 1-4.

Okhovvat, M.; Kangavari, M. R. (2019): TSLBS: A time-sensitive and load balanced scheduling approach to wireless sensor actor networks. *Computer Systems Science and Engineering*, vol. 34, no. 1, pp. 13-21.

Ren, Y. J.; Zhu, F. J.; Sharma, P. K.; Wang, T.; Wang, J. et al. (2020): Data query mechanism based on hash computing power of blockchain in internet of things. *Sensors*, vol. 20, no. 1, pp. 207.

Rostami, S. M. H; Sangaiah, A. K.; Wang, J.; Liu, X. Z. (2019): Obstacle avoidance of mobile robots using modified artificial potential field algorithm. *Eurasip Journal on Wireless Communications & Networking*, vol. 70, no. 2019.

Ren, Y. J.; Liu, Y. P.; Ji, S.; Sangaiah, A. K.; Wang, J. (2018): Incentive mechanism of data storage based on blockchain for wireless sensor networks. *Mobile Information Systems*, vol. 2018, pp. 1-10.

Ren, Y. J.; Leng, Y.; Cheng, Y. P.; Wang, J. (2019): Secure data storage based on blockchain and coding in edge computing. *Mathematical Biosciences and Engineering*, vol. 16, no. 4, pp. 1874-1892.

Ren, Y. J.; Leng, Y.; Zhu, F. J.; Wang, J.; Kim, H. J. (2019): Data storage mechanism based on blockchain with privacy protection in wireless body area network. *Sensors*, vol. 19, no. 10, pp. 2395.

Ren, Y. J.; Qi, J.; Liu, Y.; Wang, J.; Kim, G. (2020): Integrity verification mechanism of sensor data based on bilinear map accumulator. *ACM: Transactions on Internet Technology*. http://doi.org/10.1145/3380749.

Sangaiah, A. K.; Darshan, M.; Kumar, S. K.; Shamim, H. M.; Ghulam, M. et al. (2020): Blockchain enabled distributed security framework for next generation IoT: an edge-cloud and software defined network integrated approach. *IEEE Internet of Things Journal*.

Sun, W.; Zhang, X.; Zhang, X.; He, X.; Zhang, G. (2019): Driving behaviour recognition based on orientation and position deviations. *International Journal of Sensor Networks*, vol. 30, no. 3, pp. 161-171.

Singla, V.; Malav, I. K.; Kaur, J.; Kalra, S. (2019): Develop leave application using blockchain smart contract. *Conference on Communication Systems & Networks*, pp. 547-549.

Swan, M. (2015): Blockchain thinking: The brain as a decentralized autonomous corporation. *IEEE Technology & Society Magazine*, vol. 34, no. 4, pp. 41-52.

Wang, J.; Gao, Y.; Wang, K.; Sangaiah, A. K.; Lim, S. J. (2019): An affinity propagation-based self-adaptive clustering method for wireless sensor networks. *Sensors*, vol. 19, no. 11, pp. 2579.

Wang, J.; Wu, W. B.; Liao, Z. F.; Wang, L. (2019): An energy-efficient off-loading scheme for low latency in collaborative edge computing. *IEEE Access*, vol. 7, pp. 149182-149190.

Wang, J.; Gao, Y.; Zhou, C.; Sherratt, R. S.; Wang, L. (2020): Optimal coverage multi-path scheduling scheme with multiple mobile sinks for WSNs. *Computers, Materials & Continua*, vol. 62, no. 2, pp. 695-711.

Wang, C. X.; Shao, X.; Gao, Z.; Zhao, C. X.; Gao, J. (2019): Common network coding condition and traffic matching supported network coding aware routing for wireless multihop network. *International Journal of Distributed Sensor Networks*, vol. 15, pp. 1-20.

Wang, X. F.; Wang, L.; Zheng, Y. H.; Wang, J. (2018): An event-driven plan recognition algorithm based on intuitionistic fuzzy theory. *Journal of Supercomputing*, vol. 74, no. 12, pp. 6923-6938.

Wang, J.; Gu, X. J.; Liu, W.; Sangaiah, A. K.; Kim, H. J. (2019): An empower hamilton loop based data collection algorithm with mobile agent for WSNs. *Human-centric Computing and Information Sciences*, vol. 9, no. 1, pp. 1-14.

Wang, J.; Wu, W. B.; Liao, Z. F.; Sherratt, R. S.; Kim, G. J. et al. (2020): A probability preferred priori offloading mechanism in mobile edge computing. *IEEE Access*, vol. 8, no. 1, pp. 39758-39767.

Wang, J.; Yang, Y. Q.; Wang, T.; Sherratt, R. S.; Zhang, J. Y. (2020): Big data service architecture: A survey. *Journal of Internet Technology*, vol. 21, no. 2, pp. 393-405.

Wan, W. N.; Chen, J.; Zhang, S. B. (2019): A cluster correlation power analysis against double blinding exponentiation. *Journal of Information Security and Applications*, vol. 48, pp. 1-8.

Xia, Y. H.; Ying, C.; Lin, G. F.; Sun, Z. X. (2019): A third-party mobile payment scheme based on NTRU against quantum attacks. *IEEE Access*, vol. 7, pp. 56070-56080.

Xu, J.; Zhang, Y. J.; Fu, K. Y.; Peng, S. (2019): SGX-based secure indexing system. *IEEE Access*, vol. 7, pp. 77923-77931.

Zhou, Q. Y.; Luo J. J. (2018): The study on evaluation method of urban network security in the big data era. *Intelligent Automation and Soft Computing*, vol. 24, no. 1, pp. 133-138.

Zhang, X.; Duan, J.; Sun, W.; Jha, S. (2019): A tumour perception system based on a multi-layer mass-spring model. *International Journal of Sensor Networks*, vol. 31, no. 1, pp. 24-32.

Zhou, Y.; Liu, T.; Tang, F.; Tinashe, M. (2019): An unlinkable authentication scheme for distributed IoT application. *IEEE Access*, vol. 7, pp. 14757-14766.

Zhao, X. J.; Zhang, X. H.; Wang, P.; Chen, S. L.; Sun, Z. X. (2018): A weighted frequent itemset mining algorithm for intelligent decision in smart systems. *IEEE Access*, vol. 6, pp. 29271-29282.

Zhang, J. Y.; Zhong, S. Q.; Wang, T.; Chao, H. C.; Wang, J. (2020): Blockchain-based systems and applications: A survey. *Journal of Internet Technology*, vol. 21, no. 1, pp. 1-14.

Zhang, W. Y.; Shi, F. Y.; Hu, S. B.; Jian, M. W. (2018): A visual secret sharing scheme based on improved local binary pattern. *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 32, no. 6.