Received: 3 April 2021 | Revised: 24 August 2021 | Accepted: 4 September 2021

DOI: 10.1002/int.22676

RESEARCH ARTICLE

WILEY

EviChain: A scalable blockchain for accountable intelligent surveillance systems

Jiaping Yu[1] | **Haiwen Chen**[1] | **Kui Wu**[2] | **Tongqing Zhou**[1] | **Zhiping Cai**[1] | **Fang Liu**[3]

[1]Department of Computer Science, College of Computer, National University of Defense Technology, Changsha, China

[2]Department of Computer Science, University of Victoria, Victoria, Canada

[3]School of Design, Hunan University, Changsha, China

Correspondence
Tongqing Zhou and Zhiping Cai, College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China.
Email: zhoutongqing@nudt.edu.cn and zpcai@nudt.edu.cn

Abstract

Smart cameras, as typical IoT devices, are widely adopted to provide surveillance on individuals, homes, and the environment. The unavoidably captured sensitive visuals via these cameras may raise significant security concerns, while the prevalent software defects and authentication misconfiguration issues aggravate the vulnerability of such devices. However, traditional cryptography techniques are inadequate to provide full protection of these devices due to the large computation overhead. In this context, realizing accountability for these surveillance systems shall be the last line of defense in the presence of fast-evolving and high-influential threats. We propose EviChain, a scalable blockchain-based solution to trace the operations on intelligent surveillance cameras and reserve the evidence for any misuse in tamper-proofing manipulation records. Building a blockchain over the distributed cameras is challenging due to the limited capacity of on-board memory. To tackle this challenge, we design a cooperative mechanism that enables cameras to adaptively join in groups and share storage for recording blocks. In addition, we present a computation efficiency and delay-aware block generation strategy to reduce the cost of the consensus process. We perform extensive

Jiaping Yu and Haiwen Chen contributed equally to this study.

simulations to validate the superior performance of EviChain over other baselines, for example, Practical Byzantine Fault Tolerance (PBFT).

KEYWORDS
accountability, blockchain, IoT security, smart cameras

1 | INTRODUCTION

Intelligent surveillance systems (ISSs), with the continuous development of computer vision, Internet of Things and semiconductor technology, have significantly blended the boundary between the physical and digital worlds through varied kinds of smart cameras.[1] Smart cameras, as typical IoT devices enabled with on-board sensing, processing and communication, are deployed along roads, in malls, at homes, and so forth. They work as the basis of many applications, such as environmental monitoring, smart homes, and elderly care.[2] For example, the Nest Cam* and the TP-Link Kasa Cam† have become ubiquitous for surveillance tasks in both law enforcement (e.g., evidence collection) and residential communities.

The security and privacy issues of ISSs are the major concerns as the sensitive information of a large group of people would be captured and recorded.[3] When interacting with ISSs, adversaries can misuse the distributed cameras (e.g., monitoring home environments based on remote control) with legitimate identity, and can even stealthily view critical infrastructures (e.g., military sites) with minimal effort.[4] With the continuous disclosure of vulnerability, it has been revealed that attacks targeting the resource-constrained webcams have spiked in the past few years,[5] represented by the notorious Mirai that getting over 0.4 million devices compromised.

To make things worse, thoroughly protecting the ISSs from the threat of misbehavior is extremely challenging in practice, largely due to the weak security practice and resource limitations of smart cameras.[6] On one hand, though password-based method is straightforward and convenient to implement,[7] the configuration vulnerabilities of default login credentials and weak password are prevalent, making the authentication bypassed easily. Meanwhile, smart camera vendors have been criticized to have chronic neglect in applying even basic security practice,[4] which directly results in the proliferation of Mirai and its mutations. On the other hand, traditional cryptography techniques for data confidentiality are computation-intensive or requiring specific modification of protocol,[8] thus are usually not adopted on the low-performance cameras. Overall, it is not uncommon that an attacker can successfully compromise the authentication mechanism and enter the current ISSs.

In view of the misuse threats under legitimate camouflage, we argue that providing accountability for the camera system should be the last resort for the administrators. Generally, an accountable (tractable) system can utilize the access and operation records to store evidence, uncover risks, and aid later investigations.[9] Furthermore, we can attain non-repudiation on the interactions by assuring that the records cannot be deleted. In this way, misuse can be intimidated as adversaries are warned for misbehavior.

Researchers have proposed centralized solutions to enhance the system transparency and accountability. But most of these mechanisms are only applicable in the scenario where we could assume a secure and trustable centralized authority like electronic voting and payment. These kind of assumptions are too casual in surveillance system, especially in a real-world

scenario. First, not all surveillance networks are controlled by trustable authorities. When facing accidents, the responsible person may tend to modify the surveillance data to escape punishment. For example[‡]: on Feb. 2021, a trucking owner in Woonsocket, Rhode Island sentenced for falsifying electronic logs following a fatal crash. Besides, even with trusted authorities, a dishonest employee with permission could also make destructive changes to the recorded data. According to Euro Weekly News[§]: Two correctional officers at the prison where Jeffrey Epstein was held finally Admit To Falsifying Log Records on the night in August 2019, when Epstein committed suicide in his cell. Though working in a trustable authority, it is hard to avoid dishonest personnel making malicious modifications to the surveillance data. To the best of our knowledge, no previous works can address these side-effects in a centralized solution. By nature, the blockchain is accountable (or traceable) as records on the chain are validated together by all the peers.[10] Following this idea, we integrate the blockchain technologies into the ISSs to realize accountability during run-time.

We propose a novel blockchain model, *EviChain*, that records the manipulation log as transactions on smart cameras with tamper resistance. By monitoring and recording the data operations (read, write, and delete), *EviChain* can provide evidence for the runtime operations, which enables the detection, tracing and identification of any misuse. Such an non-repudiation property can further force the adversaries to meet obligation and thwart the potential malicious behavior. Nevertheless, it is nontrivial to design and implement *EviChain* in practice, due to the following difficulties:

- Limited memory resources: The available memory on typical smart cameras is generally limited.[11] Let each node to store the log information of the whole system may easily exhaust the memory spaces, especially when the number of devices is large. As a result, we need to carefully weigh the trade-off between the performance of cameras and the security requirements.
- Computation intensive consensus process: The computation and communication burden to achieve consensus among the cameras can be heavy. Video data processing is of the utmost priority in computation resources allocation, so frequent block generation and consensus request are inappropriate. Meanwhile, simply postponing the consensus will impact the security of the blockchain system.

To address the first difficulty, we design a cooperative block storage mechanism by dividing cameras into groups and treating each group as one virtual node in the blockchain system. For cameras in the same group, storage is shared to record only one replication of the block bodies. The size of group can be tuned to balance the memory limitation and security risk. We prove that finding an optimal grouping is NP-hard considering the diverse memory capacity of different cameras. As such, we present a greedy algorithm to find an approximate solution.

To address the second difficulty, we design a block generation algorithm by jointly considering the computation cost and the consensus delay. Briefly, the system could wait for a certain number of authenticated transactions before rendering a consensus request. Meanwhile, if the waiting time exceeds the delay boundary, request will be sent out immediately to mitigate the possible log tamper threats.

Overall, we make the following contributions:

- We investigate the misuse threats (both authorized and non-authorized) in ISSs. A novel blockchain model is proposed to ensure accountability on the access and operation of the system and thwart the misbehavior of adversaries.

- We design a camera node grouping strategy towards cooperative block storage to alleviate storage cost. The NP-hardness of finding the optimal grouping solution is proved and a greedy algorithm is adopted for attaining an approximate solution. Meanwhile, we introduce a delay-aware algorithm for efficient block generation during the consensus process.
- We analyze the security of *EviChain* against misuse threats and evaluate the performance of *EviChain* through extensive simulations. The results demonstrate the effectiveness and the superiority of our proposal over the baselines in terms of storage occupation and transmission cost.

The rest of this paper is organized as follows: Section 2 presents the related work. Section 3 describes the threat model and assumptions of this study. Section 4 presents an overview for the system. Section 5 gives a detailed description on the system design with security analysis on mitigating misuse threats. We evaluate the performance of *EviChain* in Section 6. In Section 7, we discuss the limitations of *EviChain*. Section 8 concludes the paper.

2 | RELATED WORK

Work relevant to this paper can be classified into two groups: Content Security of ISS and application of blockchain in IoT.

2.1 | Content security of ISS

The concern of ISS security increases with the widespread deployment of intelligent surveillance devices and networks. Data generated by smart cameras usually contains private information, while there is no unified policy taking the general security threats into consideration.[11–14]

Dai et al.[1] pointed out that ISSs are deployed in autonomous environments and this could result in malicious behavior, for example, adversaries may get physical access to these devices and attain private keys or personal data. Zhao et al.[15] analyzed the resilience of the q-composite key predistribution scheme, and enhanced the security of private keys of the Eschenauer–Gligor scheme in the neighbor discovery phase. These works focus specifically on random key predistribution schemes. In practice, the data stored in those devices can also be the target of the adversaries.

In conventional security research, public key encryption is commonly used to protect the privacy data. In [16], researchers designed a special oscillator-based random number generator in the SD card, which could be used for encrypting surveillance data. This random number generator is low-cost and can save energy consumption. In [8], Jia et al. argued that the conventional encryption method cannot protect the data security if the adversaries can get access to the keys used to encrypt the data. They thus redesigned the Flash Translation Layer for devices to achieve deniability and to eliminate the deniability compromises from NAND flash.

Due to the limited storage and computing resources of smart cameras, most conventional security mechanisms, including frequency hopping communication and public-key encryption, cannot be deployed to the intelligent surveillance network directly.[17] To make things worse, the adversaries can make use of these limitations. For example, Low et al.[18] mentioned that adversaries may drain the energy of smart cameras and other IoT devices by keep sending corrupted data. Angelo et al.[19] pointed out that the attackers can take advantage of the

inadequate authentication mechanism so that the spoofed malicious devices can be appended to the network.

It is hard to prevent all the threats attempting to make use of the resource limitation of IoT devices. In this case, the best that we can do is to make every access accountable so that the culprit can be easily identified when malicious behavior is identified. Manufacturers have realized the importance of accountability, and most of the IoT devices mandate the operation logging. Yet, according to Ho et al.,[20] attackers can get rid of the security mechanism and modify the logging procedures. Therefore, it is necessary to develop an operation logging process that is tamper resistant.

2.2 | The application of blockchain in IoT

Recently, blockchain is adopted as one of the most promising technologies to provide security support for IoT systems.[21] It was initially applied to provide digital payments,[22] and is now commonly used in smart contracts[23,24] and data storage.[25] Gueta et al.[26] developed a scalable blockchain system based on optimized Practical Byzantine Fault Tolerance (PBFT). For industrial Internet of Things, Wang et al.[27] designed a blockchain protocol for secure metering systems. With hardware assistance, Liu et al.[28] improved the scalability of the Byzantine consensus. However, the resource constraint of IoT devices and the scalability limitation of blockchain still hinder these techniques from wide adoption in IoT systems.

To address the above issues, some researchers proposed to use sidechains and additional storage devices.[29,30] For example, Misra et al.[31] deployed smart contracts on the edge and connect the devices to the blockchain to extend complex security mechanisms to those resource-constrained devices. Hossein et al.[10] developed an IoT data storage network based on cloud and the public-chain. Xu et al.[32] developed a data sharing framework, which can be applied to resource limited edges and IoT systems. Wang et al.[33] designed a three-layer structure for resource limited systems like ISSs and IoTs. In their system, sensor nodes only store transactions temporarily, the more powerful nodes like gateways contain part of the blockchain, and the full chain is stored in the multi-cloud. Since the coupling of blockchain and additional storage mechanisms may significantly increase the complexity of the system, it is necessary to provide a scalable blockchain structure for data recording without the involvement of additional infrastructures.

3 | THREAT MODEL

We attempt to build a scalable surveillance system that uses blockchain to provide accountability. Accountability here means that one can trace the operation history of all the users and cameras during the run-time.

To adopt the blockchain to storage limited devices, we propose a cooperate block storage strategy so that several, instead of one, devices can work as one virtual node to store one complete copy of the blockchain. In *EviChain*, we use a private server called *Access Gateway* to schedule the cooperate block storage strategy. It stores the device properties like ID, storage capacity, network delay, and the group ID. Here, we assume the data can only be retrieved and viewed with Access Gateway. Besides the Access Gateway, there are surveillance cameras that store the surveillance data and the blockchain which contains operation logs.

We assume that the attackers may deceive the authentication mechanism and get the valid video content stored in the camera system. In particular, we have the following assumptions about attackers:

- The attackers may cheat the security mechanism of the *access gateway* to become authorized users. In this way, the attackers may obtain data content via the *access gateway*. In other words, we do not care about who is behind authorized users and their intent, but instead, only make sure their visits can be traced back.
- The attackers, however, are not powerful enough to break into the system kernel of the *access gateway*, or block the network connections between the *access gateway* and all the cameras.

These assumptions are based on the following observations in ISSs: (i) due to weak protection, surveillance devices such as cameras are more likely to be compromised by attackers,[4,5] who may get the authorization through various methods such as dictionary attack and social engineering; (ii) the *access gateway* is much more powerful in resource than cameras and can be deployed in law enforcement or a well-protected location. As such stronger security primitives can be enforced on *access gateway* to defend against cyber attacks that compromise its kernel, and they have enough physical protection against hardware disruption.

Note that the introduction of an *access gateway* could not impact the decentralized profile, because the problem we want to address with the decentralization mechanism here is the malicious behaviors conducted through the *access gateway*, and the malicious users have no right to intervene the log distribution process. According to our assumption, malicious users can neither modify or delete the manipulation logs through the gateway nor break into the system kernel of *access gateway*.

Based on the above assumption, we mainly focus on mitigating the following malicious behavior (i.e., misuse):

1. Illegal access: Since the strength of cryptosystems depends on the designed algorithms, the resource limitations may hinder the ISSs from applying advanced encryption techniques. As a result, the adversaries may bypass the authorization mechanism and get access to the private data in the smart cameras and ruin the confidentiality.
2. Illegitimate control: Most of the ISSs do not require a strong password and tend to grant high permission to the users. The attackers could make use of these devices to threaten the ISSs and even launch a DDos attack that impacts the Internet.[5]
3. Malicious operations: Since the ISSs cannot determine the intent of an authorized user for visiting the systems, attackers with authorization may steal, or even delete, ISSs data for personal purposes. This threat is hard to detect and cannot be defeated with the traditional authentication-based or cryptography-based approaches.

4 | SYSTEM OVERVIEW

In this section, we present the architecture of *EviChain* with blockchain layer and storage layer as two building blocks. Based on the architecture design, we depict *EviChain*'s basic workflow in handling operation and data access requests.

4.1 | Architecture of *EviChain*

To adopt the blockchain in storage limited devices, we propose a cooperate block storage strategy so that several, instead of one, devices can work as one virtual node to store one complete copy of the blockchain. In *EviChain*, a private server, called *access gateway*, is used to schedule the cooperative block storage strategy and verify the users' requests. It stores the device meta information, such as device ID, storage capacity, network delay, and the group ID. Here, we assume that the data stored in surveillance cameras can only be retrieved and viewed through the *access gateway*.

Based on the existing peer-to-peer network established by the manufacturers, our system can organize the cameras and provides P2P routing and encryption communication to support specific tasks of the cameras. The mechanism needed to support the Evichain can be used directly or improved from the existing infrastructure. For example, the encrypted transmission of operation logs, the maintenance of the consensus mechanism of proposed blockchains, transaction initiation, and the PKI instantiate. Figure 1 depicts the overall architecture of *EviChain*. The core communication protocols of the proposed video surveillance systems are private P2P communication protocols based on UDP via the Internet. This kind of communication protocol has been widely deployed in most smart cameras. The architecture of camera network is logically divided into two layers: the blockchain layer and the storage layer. The blockchain layer constructs a blockchain for the system and stores the blockchain in the cameras. The storage layer is based on the storage space of all the cameras. On one hand, it stores the surveillance data captured by the cameras. On the other hand, the storage layer is designed as a decentralized file storage

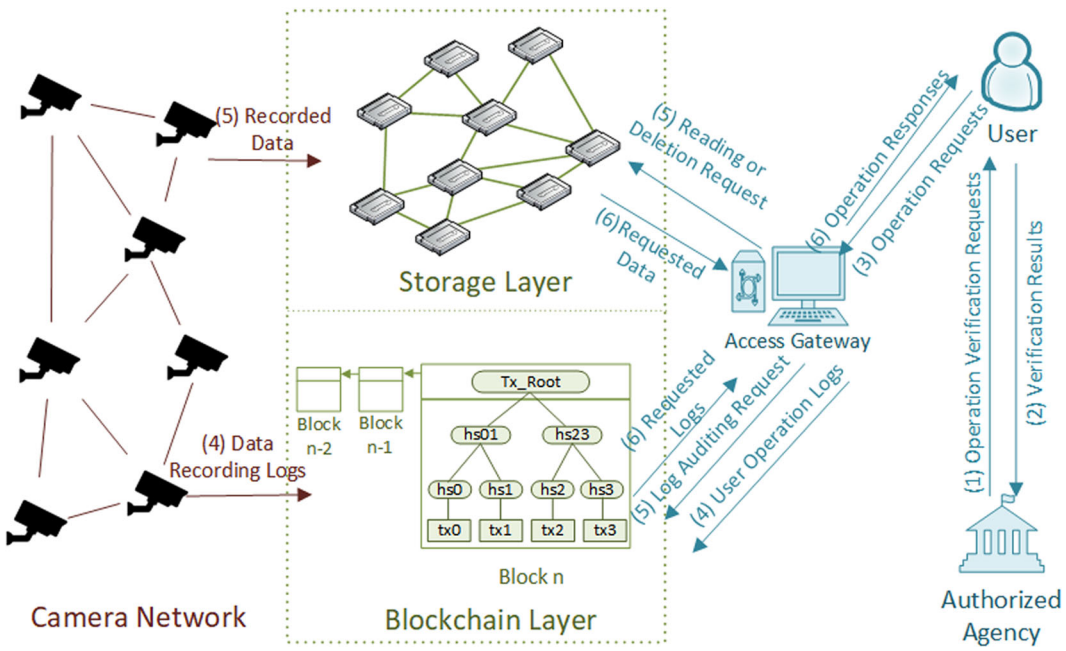

FIGURE 1 Architecture and service model of *EviChain*. This figure also briefly illustrates the workflow of *EviChain*. Authorized users first send their requests to the *access gateway*. The gateway then uploads their operation log to the blockchain and processes their requests after verification [Color figure can be viewed at wileyonlinelibrary.com]

system for *EviChain* with mature storage services like IPFS.[34] Since the Blockchain layer is the core of *EviChain*, we skip the detail of the storage layer.[35]

We assume that only the authorized users can have access to the log data stored in the blockchain[¶]. The structure of a single block is shown in Figure 2.

In *EviChain*, the transaction is used to record the operation log for data upload, retrieval, and deletion. With the hash of the current block header, a camera can determine which blocks the transaction belongs to. The public key of the user is used to verify the user's signature, which will be discussed in the next section. The hash of the operated data can be used to track the corresponding surveillance data and verify the integrity.

To implement *EviChain* in resource-limited devices and make full use of their storage space, we divide the cameras into multiple groups according to the storage space, network delay, security requirement, and the predicted number of blocks the system will generate in a certain period. Each group of cameras can be considered as a virtual node. They share their storage space and store one reputation of all block bodies (more details in Section 5.1).

4.2 | The main workflow

In principle, it is hard to determine the intention of users if they are authorized. All we can do is to record all their behaviors for future forensics. In *EviChain*, with permission from an authorized agency (such as law enforcement), users can read or delete the data stored in the storage layer or the log data stored in the blockchain. The surveillance devices can generate new data and place the data in the storage layer. All of these operations are recorded in the blockchain. As shown in Figure 1, the main workflow of *EviChain* consists of the following steps:

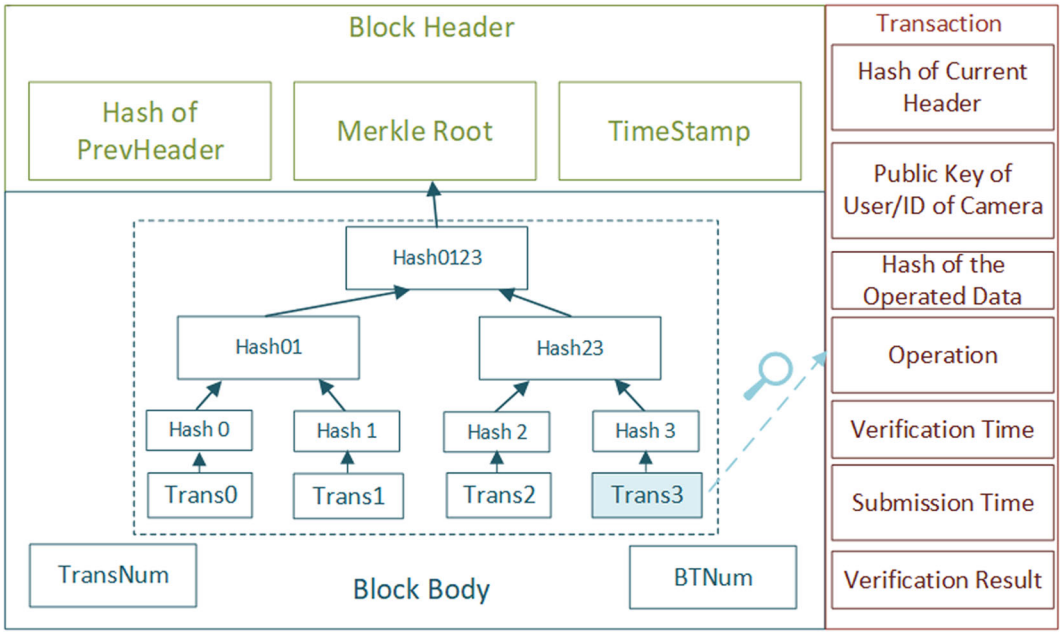

FIGURE 2 An illustrative example of a verified block in *EviChain* [Color figure can be viewed at wileyonlinelibrary.com]

1. Users need to package the operation that they want to perform (such as reading a record) and send the message with the signature signed by themselves (Sig_u) to an authority for verification.
2. If the authority passes the verification, then the authority will sign the signature again ($Sig_{ao}(Sig_u)$) and sent it back to the user.
3. The gateway verifies the two signatures signed in the previous steps.
4. After the verification is passed, the gateway uploads the operation log to the blockchain. Note that all the cameras are permanently authorized to upload the surveillance data to the storage module. Their data uploading logs are sent to the blockchain directly.
5. Once the operation log has been uploaded to the blockchain, the gateway and cameras send the request and data to the corresponding layers.
6. *Evichain* sends the requested data and operation result back to the gateway. Then the gateway sends them back to the user.

5 | DETAIL DESIGN ON EVICHAIN

In this section, we introduce the structure of the cooperative block storage mechanism and the consensus mechanism in detail, and analyze the security property of *EviChain*.

5.1 | Cooperative block storage

With the increment of blocks, the size of blockchain increases over time. This is called blockchain bloat. Because the mining process does not rely on the economic incentive, and the network environment is relatively safe, we choose to build a private chain with optimized PBFT mechanism. Inspired by Tian et al.,[36] to reduce the impact of blockchain bloat and store the blockchain in the storage limited devices, we design a cooperative block storage mechanism. The main idea of this mechanism is to divide the cameras into multiple groups and distribute the blockchain body to different cameras. Since *EviChain* runs in a secure network, and only the authorized devices can have access to the network, if we adopt a consensus mechanism that cameras do not need to frequently query other blocks, we can modify the blockchain structure to reduce the storage space occupied by the blockchain.

In *EviChain*, one or more devices in the same group are shared to record only one reputation of block bodies. To make the blockchain resilient to camera failures, we need to carefully design the camera grouping strategy such that the blockchain structure can remain stable even if one or more cameras break down.

Table 1 shows the notations used in the following context. As shown in Figure 1, we consider a system with N surveillance cameras. These cameras are divided into M groups. The Access Gateway stores all the device metadata like device ID, storage capacity, network delay, and the group ID. As shown in Figure 3, each camera stores all the block headers and a part of the block bodies. The block bodies stored in all cameras in each group are all block body in the current blockchain. Here, we define n_j as the number of devices in group j, which can be denoted as

$$n_j = \sum_{i=1}^{N} x_i^j, \tag{1}$$

TABLE 1 List of notations

Notation	Definition
N	Total number of surveillance devices
m	Total number of groups
n_i	Number of devices in Group i
x_i^j	Binary indicator, indicate if device i is in group j
C_i	Storage capacity of device i
S_h	Size of block header
S_b	Size of block body
k	Number of blocks on blockchain
r_i	Ratio of storage space occupied to the total space in device i
s_i	Size of storage space occupied in device i
$R(x)$	Security cost with x groups
$T(x)$	Transmission cost with x groups

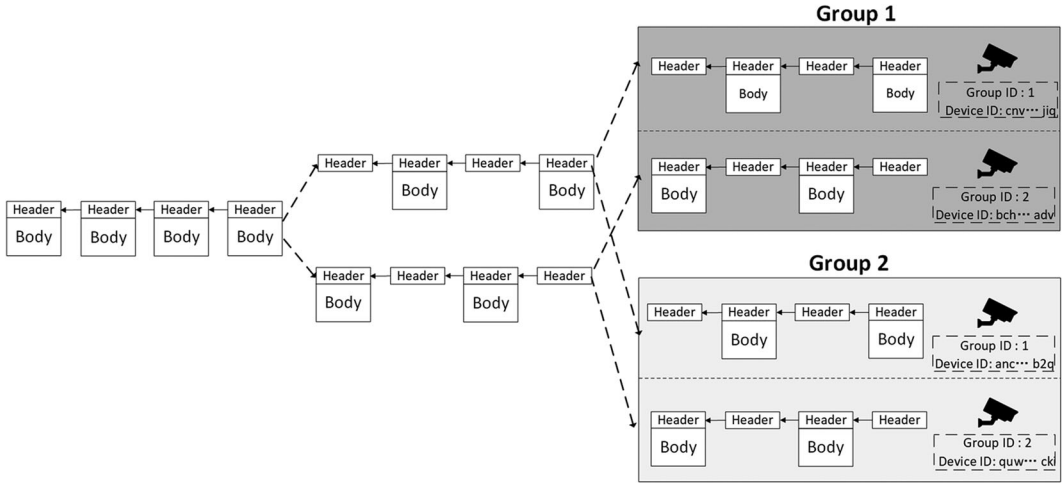

FIGURE 3 An illustrative example of cooperative blockchain storage mechanism

where x_i^j is a binary indicator. If device i is in group j, then $x_i^j = 1$. Likewise, the total number of devices in the *EviChain* network can be represented as

$$N = \sum_{j=1}^{M} n_j. \tag{2}$$

To solve the camera grouping problem, we need to not only minimize the maximum storage cost and transmission cost but also reduce the security risk. To be specific, we need to consider the following factors:

5.1.1 | The security risk

In *EviChain*, security risk means the risk of blockchain failure caused by camera failures. The worst case would be that all the cameras storing the same block in each group have failed. In this situation, the number of failed cameras that the system can withstand is the number of groups. In other words, the security risk is inversely proportional to the group number. So we calculate the security risk of *EviChain* as

$$R(m) = \frac{1}{e^m}, m \in [1, n], \tag{3}$$

where $R(m)$ represents the security risk of *EviChain*, and m represents the number of groups in the system.

5.1.2 | The storage cost

In the blockchain, the size of the ledger increases over time. However, the storage capacities of the cameras are limited. So it is crucial to minimize the size of the blockchain. The remaining space varies from device to device, and the amount of free storage space may depend on the video resolution and other factors. A camera with higher resolution may need more storage space for the proceeding surveillance data. Thus, *EviChain* not only needs to cut down the total size of the blockchain but also should minimize the impact of blockchain on the storage space of all devices. Hence, we define the storage cost as the largest proportion of storage space occupied by the blockchain among all cameras.

Specifically, the percentage of storage space occupied by the blockchain in device i can be denoted as

$$r_i = \sum_{j=1}^{M} x_i^j \times \frac{S_i}{C_i}, \tag{4}$$

where C_i is the storage capacity of device i, and S_i is the size of storage space occupied by the blockchain in device i. The storage space used by the blockchain in one camera consists of the size of all the block headers and the block bodies stored in the camera, which can be denoted as

$$s_i = (\frac{1}{n_j} \times S_b + S_h) \times k. \tag{5}$$

So the proportion of storage space occupied by the blockchain in device i is

$$r_i = \sum_{j=1}^{M} x_i^j \times \frac{(\frac{1}{n_j} \times S_b + S_h) \times k}{C_i}. \tag{6}$$

And the storage cost of group j can be denoted as

$$S_j = \max x_i^j \times \frac{(\frac{1}{n_j} \times S_b + S_h) \times k}{C_i} \quad \forall i \in [1, N]. \tag{7}$$

To minimize the impact of blockchain on the storage space of all devices, For each group, we denote the storage cost as the maximum r_i in the group.

$$S(m) = \max \sum_{j=1}^{M} x_i^j \times \frac{(\frac{1}{n_j} \times S_b + S_h) \times k}{C_i}. \tag{8}$$

5.1.3 | The transmission cost

The PBFT consensus mechanism is a communication heavy protocol, that is, the transmission cost has a significant impact on the performance of the mechanism. Here, the transmission cost could be modeled as

$$T(m) = k^2 + (m + 1) \times k - 1, \tag{9}$$

where k represents the number of blocks on the blockchain.

Overall, the camera grouping problem can be formulated as follows:

$$\begin{aligned} \min_{(m,x_i^j)} \quad & w_1 \times \max_i r_i + w_2 \times R(m) + w_3 \times T(m) \\ \text{s.t.} \quad & \sum_{j=1}^{M} x_i^j = 1, \\ & 1 \le m \le N. \end{aligned} \tag{10}$$

Here, w_1 w_2, and w_3 represents the weights of the three objects respectively.

Theorem 1. *The camera grouping problem for cooperative block storage is NP-hard.*

Proof. By allowing only instances for which $w_2 = 0$ and $w_3 = 0$, we can reduce the problem to:

$$\begin{aligned} \min_{(m,x_i^j)} \quad & \max_i \sum_{j=1}^{M} x_i^j \times \frac{(\frac{1}{n_j} \times S_b + S_h) \times k}{C_i} \\ \text{s.t.} \quad & \sum_{j=1}^{M} x_i^j = 1, \\ & 1 \le m \le N, \\ & n_j = \sum_{i=1}^{N} x_i^j \end{aligned} \tag{11}$$

for each m, the problem is equivalent to

$$\begin{aligned} \max \quad & \min_i \sum_{i=1}^{N} x_i^j \times C_i \\ \text{s.t.} \quad & \sum_{j=1}^{M} x_i^j = 1. \end{aligned} \tag{12}$$

This is an integer linear programming problem, which has been proved to be NP-hard.[37]

We adopt a greedy algorithm to solve the problem. The pseudo-code of the camera grouping strategy is shown in Algorithm 1. The main idea of this algorithm is as follows: First the algorithm traverse all the values of m and calculates the $R(m)$, $T(m)$, and $S(m)$. To calculate $S(m)$, the algorithm first creates m empty groups and sorts the storage capacity of all the cameras in increasing order. It then considers the storage capacity of these devices one at a time. Assuming the chosen device is D_i, and the storage cost of the group with only one device D_i is S_i. If there is no existing group T_j with smaller storage costs S_j than S_i, and there are still empty groups, then place the device D_i into a new group. Otherwise, place the device D_i into the group with the maximum storage cost S_j.

If two or more groups have the same storage cost, then place the device into the group with more devices. If the remaining empty group number is equivalent to the remaining device number, then assign one remaining device to each remaining group, which assures that each group has at least one device.

The camera grouping strategy needs to traverse all possible group numbers, which requires $O(N)$ times. For each possible grouping scheme, it needs to take $O(N)$ times to iterate over all devices to determine which group it should be in. Then for each device, it needs to take $O(N/2)$ times to traverse all the possible groups. Hence the time complexity of the camera grouping strategy is $O(N^3)$.

Algorithm 1 The Camera Grouping Strategy

Require: Weight w_1, w_2 and w_3.
Ensure: The number of groups m, Device grouping indicator x_i^j
1: Sort the devices based on the storage capacity in ascending order
2: *MinCost* = Infinity;
3: **for** $m = 1$ to N **do**
4: Calculate $R(m)$ and $T(m)$ in Eq. 3 and Eq. 9;
5: *EmptyGroupNum* = m;
6: **for** $i = 1$ to N **do**
7: **if** # of remaining devices = = *EmptyGroupNum* and *EmptyGroupNum*! = 0 **then**
8: **for** each empty groups **do**
9: Assign one remaining device;
10: **end for**
11: **end if**
12: **if** $S_i \leq$ the maximum storage cost of all groups or *EmptyGroupNum* = = 0 **then**
13: Place device i into the group with the maximum storage cost;
14: **else**
15: Place the device i into an empty group;

(Continues)

16: $EmptyGroupNum - -$;
17: **end if**
18: **end for**
19: $tmpCost = w_1 \times S(m) + w_2 \times R(m) + w_3 \times T(m)$;
20: **if** $tmpCost < MinCost$ **then**
21: $MinCost = tmpCost$;
22: **end if**
23: **end for**
24: Return x_i^j and m when $tmpCost == MinCost$

5.2 | The consensus process

Castro and Liskov first proposed the Practical Byzantine Fault Tolerance (PBFT) algorithm in 1999.[38] It is a state-of-practice Byzantine fault-tolerant algorithm that has been widely adopted in the consortium and private chains. The main working process of PBFT is shown in Figure 4A, where all the participatory nodes need to store and verify the newly generated blocks. This causes considerable burdens to the limited storage IoT devices in our context. As shown in Figure 4B, we adopt a modified PBFT consensus mechanism. Since we assume the ISS network is relatively safe and the storage and computation capacity of the cameras are limited, only the chosen cameras need to participate in the block verification process and store the generated full block, while the other nodes are just required to keep the block header. Generally, such modification adapts the traditional PBFT mechanism to the large-scale IoT systems. In *EviChain*, cameras are divided into several groups based on the grouping strategy introduced in the previous subsection. As shown in Figure 3, each camera has a unique device ID, and the order of creating the next block is determined by the device ID. Besides the device ID, a group ID is also assigned to cameras in each group.

When the blockchain generates a new block i, the hash value of the new block's header can be obtained by all the cameras. If we assume that the camera c is the chosen camera to generate the $i + 1$ block, then we have:

$$c = \underset{u \in V}{argmin}(H(n_u) \oplus H(b_i)), \tag{13}$$

where $\oplus$ denotes XOR operation, $H(n_u)$ is the hash value of camera u's Device ID, $H(b_i)$ is the hash value of the ith block's header, and V is the device set of *EviChain*.

In *EviChain*, each new transaction will be broadcast to the network and stored in the chosen block generator. After the camera c is chosen as the next block generator, it will count those new transactions, then the camera will generate a new block that contains a list of the unprocessed transactions.

The consensus mechanism includes four steps:

1. When a camera generates a new block, the block header will be broadcast to all cameras, and a random chosen camera in each group will get a full block, which consists of a block header and a block body.

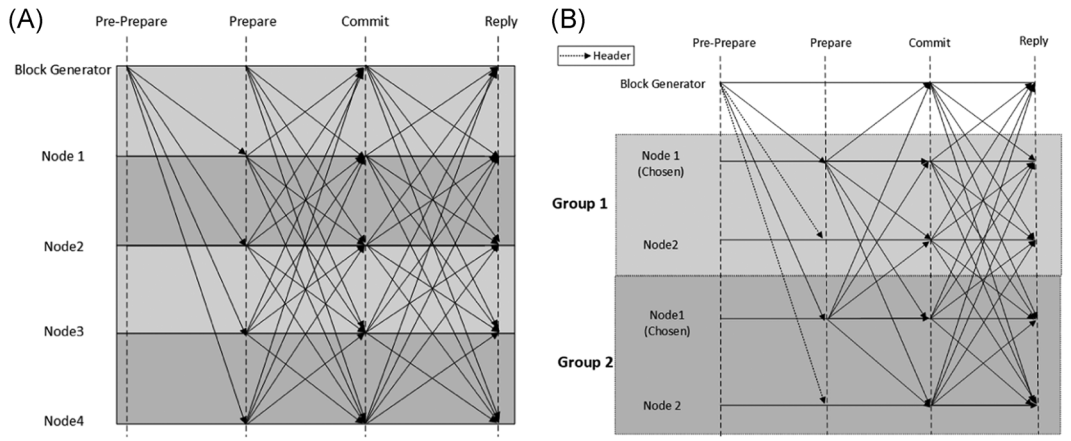

FIGURE 4 The working process of (A) the original PBFT algorithm, (B) the modified PBFT. In the modified PBFT, only the chosen nodes (i.e., Node 1 in groups 1 and 2 here) participate in block storage and verification

2. After receiving the new block, those cameras will compare whether the Merkle Root is consistent with the Merkle Root calculated by each of the transactions, and if they are consistent, broadcast the hash value of the block header to other cameras.
3. Assume that the number of groups is n, and $f = (n - 1)/3$. If a camera receives $2f + 1$ hash values of block headers that are the same as the hash value contained in the camera itself, then the camera will broadcast a confirmation message to other cameras.
4. If a camera finally receives $2f + 1$ confirmation messages, it will store the block or block header in the camera's blockchain according to the above defined rules.

In this consensus mechanism, we need to address the following critical issue: how to determine the number of transactions in each block so that the system delay is within an acceptable range?

In some blockchain systems, the number of transactions in each block is fixed. But the number of query requests in a surveillance camera system may fluctuate over time. To deal with this issue, the transaction number contained in each block should be variable. Based on this idea, we design the system to package transactions into a block if there are transactions generated after a certain time interval. This method is used in famous public chains like Bitcoin and Ethereum. However, when it comes to *EviChain*, this method should be improved to handle the special transaction patterns. In *EviChain*, most transactions are generated in some particular period of time (working hours of law enforcement, for example). If a large number of transactions occur within a certain time interval, this block would become particularly large. This not only makes the transmission cost unacceptable but also consumes a lot of computing power to pack those transactions. On the other hand, if the number of transactions is small, or there are no transactions during the generation period, empty blocks may be generated, and this wastes computing and storage resources.

To deal with this problem, we design an algorithm that automatically adjusts the consensus strategy to ensure that the delay of the system is within an acceptable range. The pseudo-code is shown in Algorithm 2.

Algorithm 2 The Block Generation Strategy

Require: Transaction Waiting time D_w, Maximum number of transactions allowed in a block Num_T
Ensure: Transactions that will be added to the chain *Trans*
1: **if** Receiving a new transaction T_1 **then**
2: *NumWaitingTrans* = 1;
3: *Trans*.add(T_1);
4: **while** true **do**
5: **if** Receiving a new Transaction T_i **then**
6: *NumWaitingTrans*++;
7: *Trans*.add(T_i);
8: **end if**
9: **if** *Timer* > D_w or *NumWaitingTrans* ≥ Num_T **then**
10: break;
11: **end if**
12: **end while**
13: **end if**
14: Return *Trans*

5.3 | Security analysis

We analyze the security of *EviChain* in terms of the threat of misuse.

First, for unauthorized adversaries who attempt to access, control the surveillance devices, or breach the sensitive data from the system, the blockchain will record the attempted operations. Such information can be used to identify the identity that being faked and remind the network manager to take measures to prevent further damage.

Second, for the misuse (either abnormal reading or illegal deletion) conducted by authorized user, operations are recorded in the system for future forensics. The recording process cannot be bypassed as the access gateway sends it to the blockchain when responding to the corresponding operation request.

Finally, once the logs are recorded by the blockchain, the adversaries cannot modify the operation logs to ruin the evidence of misbehavior, as the blockchain is tamper proof by nature.

Besides the threats of misuse, there are some other security vulnerabilities that Evichain may incur:

First, the main security vulnerability is that our cooperative block storage mechanism may weaken the robustness of Evichain, with fewer transaction replicates in the network, compared with the system with the PBFT mechanism. In Evichain, a certain block is stored in "segments" in each group, which degrades the system's robustness on Byzantine attacks. As a result, in an extreme situation where each group has a breached camera node, then this block is inaccessible from the chain. What's worse, if cameras are clustered into relatively fewer groups (e.g., 5 groups for 100 cameras) for efficiency purpose, an adversary can easily tamper the records of one cameras from half of the groups (3 cameras are enough for the 5-groups-example, instead of handling 51 cameras when no grouping are involved) to modify, or even erase, the original contents.

In fact, this type of vulnerability can be controlled with adjustable grouping scale. Intuitively, when the number of groups N_g is set to the number of camera nodes N_c, then it

becomes the raw blockchain system. We can measure the security strength of a grouping strategy i by comparing N_g^i with a task-specific threshold N_{th}. Namely, for an open environment deployment where compromising $N_{th}/2$ nodes is considered to be infeasible, the designer can set $N_g \geq N_{th}$. In this way, the Evichain is believed to provide sufficient robustness with flexible camera grouping.

Second, the underlying storage or transmission network defects may also breach the system's security. One kind of common threats in this context is that adversaries may compromise the cameras on-site and steal the on-board data. Since such threats are not launched through the gateway, Evichain cannot record their misbehavior. To provide a remedy, we point out that existing techniques on robust distributed storage can facilitate protection for on-site attacks. For example, the Centipede scheme,[35] which distribute the surveillance data across geographically dispersed cameras with geo-aware erasure coding, can be easily integrated into our Evichain to overcome such attacks. In this way, the adversaries cannot recover the surveillance data even if they manage to get full control of several devices in the network. Yet, we emphasize that this paper mainly focuses on realizing scalable accountability, and we believe the storage or transmission security issues can be mitigated with incremental mechanisms.

6 | EVALUATION

In this section, we propose numerical simulations to illustrate the performance of the proposed system. In our simulation, we deploy 10 to 100 devices in our simulated network. For each device, the storage space for blockchain is randomly generated with the volume ranging from 1 MB to 16 MB.

6.1 | Effectiveness of cooperative storage mechanism

We first evaluate the effectiveness of the proposed cooperative block storage mechanism by comparing it with the original PBFT mechanism.

In this experiment, the waiting time for generating a new block is set to 10 min. The maximum number of transactions in one block is set to 1000. To observe the details, all the blocks generated in the experiment are full blocks, namely, the corresponding number of transactions is the maximum, which means 1000 here.

Since the practical requirements on security level and limitations on storage vary under different scenarios, we present four distribution strategies for different design principles as follows:

Storage Oriented: This distribution strategy mainly focuses on saving storage space. This strategy is mainly targeted at users who are sensitive to storage costs, such as small and medium-sized enterprises. It assigned the 90% weight to storage cost, 5% to the transmission, and 5% to security.

Security Oriented: This distribution strategy pays more attention to the security of the protection system. All the cameras are assigned 90% weight to security cost, and 5% to the storage, and 5% to transmission. This strategy is mainly for high-risk scenarios where the cameras are more vulnerable to physical damage, such as law enforcement or conflict areas.

Transmission Oriented: The main purpose of this transmission strategy is to reduce the transmission cost. It assigns 90% weight to transmission cost. This strategy is for scenarios with limited bandwidth.

Balanced: Under this strategy, we strike a balance among security, storage, and transmission so that the storage space occupation, transmission delay and system security are all considered.

Finally, the topology is initialized with 100 devices, which sequentially generate 100 blocks during the simulation. Details on the simulation result are shown in Table 2. We can see from Table 2 that the storage and transmission overhead of original PBFT is much higher than the optimized distribution strategies. The *Security Oriented* and *Balanced* strategies consume more storage and transmission resources than *Transmission Oriented* and *Storage Oriented* strategies to provide better security protection. To further evaluate the properties of *EviChain*, we test its effectiveness in terms of different cost measures in the following experiments.

Note that to evaluate the impact of both network size and network load, in this section, each sub-experiment is tested separately in two scenarios: one is to test different load in a network of fix size, the other is to deliver the same load in the networks of different sizes.

6.1.1 | Storage cost

We first conduct the fixed size experiment. The network generates a different number of blocks on a system with 100 devices to evaluate the amount of storage space occupied by the blockchain layer. Then we counted the size of blockchain in each camera.

Figure 5A shows the block size variation with the growth of block number. The storage overhead is measured based on the biggest fraction of the blockchain occupied in all cameras. As we can see, with the increment of block numbers, the storage occupation ratio of the original PBFT strategy increases linearly, to the Cooperative Storage Mechanism, the storage occupation ratio rising volatility. That is because the block generator is randomly chosen based on the hash value of the previous block header (as shown in Equation 13).

In Figure 5A, the storage space occupied by the traditional PBFT distribution strategy reaches 50% when the system generated 100 blocks, which is unacceptable in practical deployment. The fractions of Cooperative Block Storage Mechanisms are much lower than the traditional PBFT. Among them, the *Security Oriented* and *Balanced* strategies are higher than *Transmission Oriented* and *Storage Oriented* strategies. That is because to assure the robustness of *EviChain*, each camera needs to share more storage space to store the blockchain. With

TABLE 2 Evaluation result of generating 100 blocks from 100 devices

	Original PBFT	Storage Oriented	Security Oriented	Transmission Oriented	Balanced
Group number	–	2	9	2	5
Message number	2,141,501	1,161,501	1,231,501	1,161,501	1,201,501
Largest blockchain size (KB)	10,400	1000	2500	1100	2000
Largest fraction	49.61%	3.82%	7.63%	4.77%	5.72%

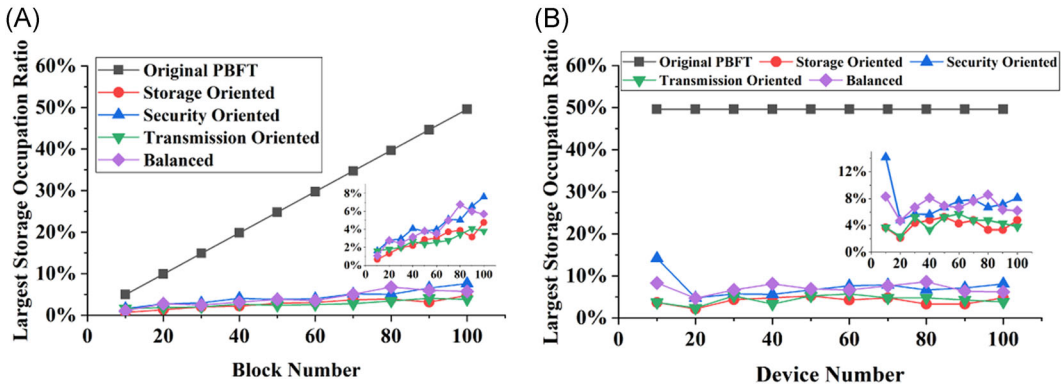

FIGURE 5 (A) The blockchain size with the growth of block number under traditional PBFT and four distribution strategies. (B) The largest fraction of storage space occupation in all devices [Color figure can be viewed at wileyonlinelibrary.com]

Storage Oriented strategy, it can significantly cut down the size of the blockchain in each camera by dividing the cameras into fewer groups than *Security Oriented* and *Balanced* strategies. Since the *Transmission Oriented* strategy also needs to reduce the group number to reduce the transmission cost, its storage cost is in the same range as *Storage Oriented* strategy.

Then we further deploy the fix load simulation to evaluate the impact of network size on the system storage cost. In this simulation, we distribute 100 blocks in the network with the size from 10 to 100 devices, and the result is shown in Figure 5B.

According to the result, the *Transmission Oriented* strategy and the *Balanced* strategy need more storage spaces than *Transmission Oriented* and *Storage Oriented* strategies. But the size of blockchain in all the four optimized distribution strategies are all much smaller than traditional PBFT.

6.1.2 | Transmission cost

Note that it is not easy to measure the network load, communication latency, or other parameters in a simulation precisely. To measure the transmission cost, we compare the message number and message size of traditional PBFT and the modified cooperative block storage mechanism. Here, message number means the total number of messages transmitted during the consensus process, and message size means the total size of all messages received by all the devices.

To evaluate the impact of different loads on the transmission cost, we first conduct three fixed-size experiments in networks with 20, 50, and 100 devices. As shown in Figure 6, the message number increases with the increment of block numbers. The results of modified cooperative storage mechanisms are much lower than the traditional PBFT. The message number of *Security Oriented* strategy is higher than the other modified strategies. The *Balanced* strategy is lower than the *Security Oriented* strategy but higher than *Storage Oriented* and *Transmission Oriented* strategies. The *Storage* and *Transmission Oriented* strategies are in the same level.

The fixed load experiment shows the same result. Figure 7 shows the total message size transmitted during the consensus process when generating 20, 50, and 100 blocks respectively. The traditional PBFT is much higher than those of the modified cooperative storage mechanism in message size. And the message size of the security oriented strategy is larger

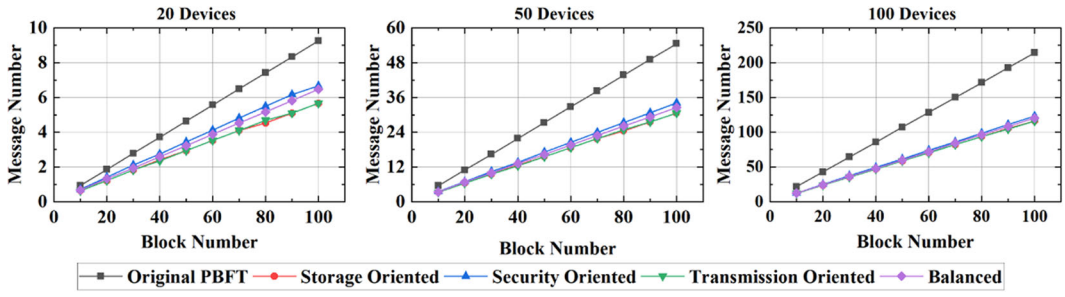

FIGURE 6 The message overhead of PBFT and the modified cooperative block storage mechanism in the network with 20, 50, and 100 devices [Color figure can be viewed at wileyonlinelibrary.com]

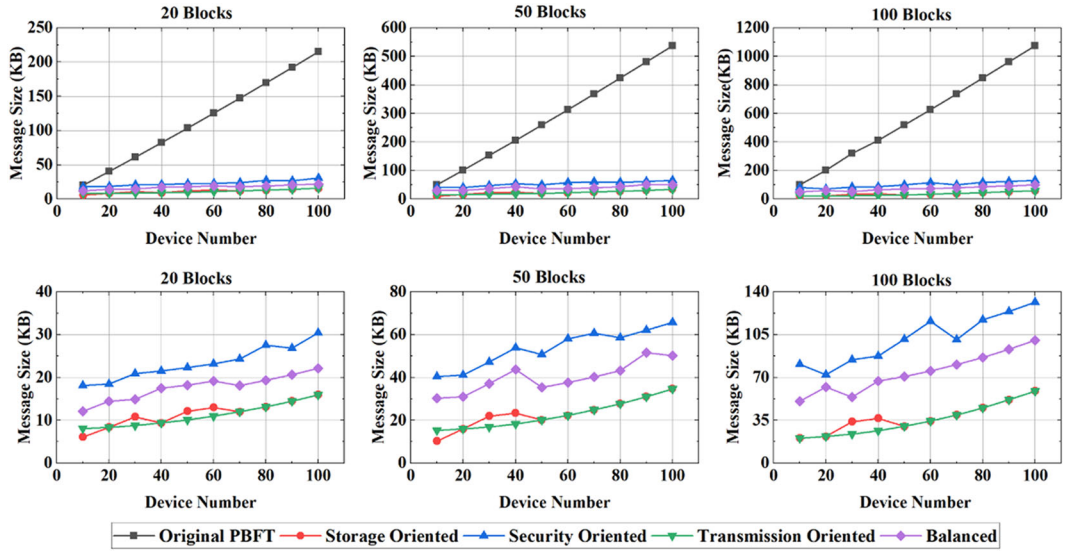

FIGURE 7 The message size transmitted in the network during the generation of 20, 50, and 100 blocks [Color figure can be viewed at wileyonlinelibrary.com]

than the other three distribution strategies. Transmission Oriented and Storage Oriented has relatively the lowest transmission cost.

6.1.3 | Security cost

We evaluate the robustness of the four camera grouping strategies.

To simulate the unauthorized adversaries who attempt to destroy cameras and evaluate the security of the system, we randomly selected cameras to go offline until the blockchain crash. Here, crash means part of the blockchain data cannot be found and recovered from the network. We conduct 10, 000 runs for each strategy. The result of the fixed size scenario is shown in Figure 8A.

From Figure 8A, we can see that *Storage Oriented* and *Transmission Oriented* strategies can barely survive even only a few devices are disabled. Though it still has a chance to survive even if nine cameras crashed, in most of the simulation, the system crashed when three devices are disabled.

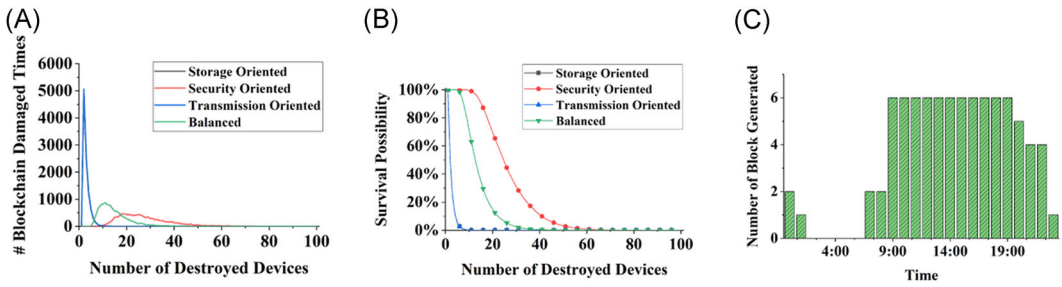

FIGURE 8 (A) The number of times the blockchain layer crash with the growth of the number of destroyed cameras. (B) The survival possibility of different distribution strategies when the system under attack. (C) The number of blocks generated at each time [Color figure can be viewed at wileyonlinelibrary.com]

For the *Balanced* strategy, the system can tolerate no more than five cameras crashed. The system is most likely to crash when there are 10 devices failed. It still has a chance to survive even after 30 cameras are destroyed.

For the *Security Oriented* strategy, the system can survive if less than 10 cameras are disabled. It is most likely to crash when 20 devices are disabled. In some cases, the system can still survive even 60 devices crash.

Figure 8B shows the survival possibility when a different number of devices are disabled. The system with *Storage Oriented* and *Transmission Oriented* strategies could barely survive even if only three to four devices fail. And when the number of disabled cameras reaches 10, the system completely failed.

For the *Balanced* strategy, the system could tolerate more than 10% failed cameras. When attackers destroy 15 out of 100 cameras, *EviChain* still have half the chance to survive. As long as the number of cameras destroyed is less than 30, the system still has a chance to survive.

The *Robust Oriented* strategy have the best performance in this test. It can tolerate more than 20 of failed cameras. When the number of failed cameras reach 25, it can still assure more than 50% survival rate.

In a fixed load scenario, we simulate the attack launch to systems with 10 to 100 cameras. Since the device number of the *EviChain* has a more significant impact on the result of the security analysis experiment, we show a more detailed result in Table 3. It shows the group number of each network and the average number of destroyed devices when blockchain crashed.

We denote the number of destroyed devices when blockchain crashed as D_n. According to Table 3, for all the four distribution strategies, both the group number and the average number of destroyed devices the system can withstand reach the top when the device number is among 30 to 60. That is because when the device number is more than 60, the storage and transmission costs have more effect on the node distribution.

6.2 | Results on block generation

In this part, we simulate the number of blocks generated in 24 h to analyze the performance of the *Block Generation Strategy*. We virtually deploy 10 access computers to simulate the real usage of the system and the time interval is set to 10 min. For each time interval, we randomly generate the number of transactions to simulate the random access during each time period. In

TABLE 3 Simulation results of security cost with different network size

Device number	Storage Oriented		Security Oriented		Transmission Oriented		Balanced	
	Group number	D_n	Group number	D_n	Group number	D_n	Group number	D_n
10	2	1	8	8	3	3	6	7
20	3	4	9	13	3	4	7	11
30	4	11	10	20	3	9	7	17
40	4	9	10	26	3	5	8	24
50	3	4	11	32	3	4	8	26
60	3	4	11	37	3	4	8	28
70	3	4	10	31	3	4	8	23
80	3	4	11	34	3	4	9	27
90	3	4	12	38	3	4	8	22
100	3	4	11	35	3	4	8	22

Note: D_n: the number of destroyed devices when blockchain crashed.

practice, less retrieval or deletion requests are conducted at night, and since most of the surveillance camera records only when motion is detected, there are fewer surveillance data that will be generated. So the upper and lower bound of the transaction generation process at night is much lower than in the daytime. The result is shown in Figure 8C.

As we can see, the number of blocks generated at night is fewer than during the day. The system worked with its full capacity from 8 a.m. to 7 p.m. After 8 pm, the number of transactions decreased, and the system could cut down the number of generated blocks to avoid wasting computing power. From 2 a.m. to 6 a.m., the block generation completely halted because no user attempted to retrieve surveillance video, and no camera captured any motion during that period.

7 | DISCUSSION

Based on our experiments and evaluation, we can conclude that:

- With the blockchain layer, the system can track the access records of all the authorized users. This is extremely useful for forensic purposes.
- *EviChain* greatly reduces the size of blockchain while ensuring accountability and robustness, so that it can be implemented in cameras with limited storage space. This feature enhances the scalability of the system.

In addition, the current version of *EviChain* has some limitations that deserve future investigation:

- We assume that the data and communication security could be assured by existing techniques.[39,40] Yet, how to maintain the efficiency of the system when adopting

cryptography methods is nontrivial, especially for the interaction process of different devices in the blockchain layer.

- While we have designed a scalable solution to alleviate blockchain bloat, the blockchain layer may still fail if a large number of cameras were disabled. The only way to avoid this problem is to make all cameras to be the full node. Here, full node means the camera stores all the block header and block body. This would significantly increase the storage cost and reduce the scalability of *EviChain*. In practice, one can gradually increase the number of camera groups to avoid the threats incurred by large amount of abnormal nodes.
- As for the choice of the consensus algorithm, we agree that, with an efficient consensus mechanism, like RAFT,[41] the efficiency can be further improved. This study is devoted to the security of pervasive IoT cameras, so security and scalability are on the priority of our design, which can be well satisfied with PBFT. As we know, RAFT can only be deployed in a private chain. Considering that the working scenarios of the surveillance systems can be complicated, a consortium chain should be an essential choice during the deployment of Evichain. Based on these considerations, we adopt the PBFT mechanism in the design. In the future, we could further improve the system performance by applying more efficient consensus mechanisms.

8 | CONCLUSION

This study focuses on mitigating the threats of potential misbehavior, especially from authorized users, in ISSs. We state that realizing accountability is key to thwarting the threats as misbehavior would be exposed if users are not align with the obligation. We present a scalable blockchain-based system to record the operations on cameras and thus reserve evidence for any misuse. In detail, we design a novel block storage mechanism by camera grouping to cooperatively utilize the cameras' storage. We also introduce an adaptive block generation strategy in the consensus process for computation efficiency. Extensive simulations have been done and the results demonstrate the effectiveness and superior performance over PBFT.

In the future, we plan to develop a dynamic camera grouping strategy considering the involvement of mobile cameras (e.g., launched on a vehicle) and test the performance of EviChain with real-world deployment.

ACKNOWLEDGMENTS

This study is supported by the National Natural Science Foundation of China (62072465, 62102425), the National Key Research and Development Program of China (2020YFC2003400, 2018YFB0204301), the Hunan Provincial Postdoctoral Innovation Talent support Program (No. 2021RC2071), and the NUDT Research Grants (No. ZK19-38).

ENDNOTES

*Nest Cam Indoor https://store.google.com/us/product/nest-cam

†Kasa Smart Security Cameras—Kasa Smart https://www.kasasmart.com/us/products/security-cameras

‡https://www.freightwaves.com/news/trucking-owner-sentenced-for-falsifying-eld-logs-following-fatal-crash

§https://www.euroweeklynews.com/2021/05/22/jeffrey-epstein-prison-guards-admit-to-falsifying-log-records/

¶Note that the malicious behavior of an authorized user still exist under such an assumption, since our threat model assumes that an attacker may compromise the authorization.

ORCID

Jiaping Yu https://orcid.org/0000-0003-4367-3179
Haiwen Chen https://orcid.org/0000-0001-8031-8008
Kui Wu https://orcid.org/0000-0002-2069-0032
Tongqing Zhou https://orcid.org/0000-0002-6620-1898
Zhiping Cai https://orcid.org/0000-0001-5726-833X
Fang Liu https://orcid.org/0000-0001-8753-3878

REFERENCES

1. Dai H, Zheng Z, Zhang Y. Blockchain for internet of things: a survey. *IEEE Internet Things J*. 2019;6: 8076-8094.
2. Mora H, Gil D, Terol R, López J, Szymanski J. An IoT-based computational framework for healthcare monitoring in mobile environments. *Sensors*. 2017;17:2302.
3. Lin J, Yu W, Zhang N, Yang X, Zhang H, Zhao W. A survey on internet of things: architecture, enabling technologies, security and privacy, and applications. *IEEE Internet Things J*. 2017;4:1125-1142.
4. Ioannis S, Panayiotis K, Mihalis P, Cristina A, Javier L. A survey of IoT-enabled cyberattacks: assessing attack paths to critical infrastructures and services. *IEEE Commun Surv Tutor*. 2018;PP(4):1.
5. Constantinos K, Georgios K, Angelos S, Jeffrey V. DDoS in the IoT: Mirai and other botnets. *Computer*. 2017;50(7):80-84.
6. Wu H, Zhang J, Cai Z, et al. Resolving multi-task competition for constrained resources in dispersed computing: a bilateral matching game. *IEEE Internet Things J*. 2021. http://doi.org/10.1109/jiot.2021.3075673
7. Sood SK, Sarje AK, Singh K. Cryptanalysis of password authentication schemes: current status and key issues. IEEE. 2009:1-7.
8. Jia S, Xia L, Chen B, Liu P. DEFTL: Implementing plausibly deniable encryption in flash translation layer. CCS '17 Association for Computing Machinery; New York, NY, USA. 2017:2217-2229.
9. Singh J, Millard C, Reed C, Cobbe J, Crowcroft J. Accountability in the IoT: systems, Law, and Ways Forward. *Computer*. 2018;51:54-65.
10. Hossein S, Lukas B, Anwar H, Simon D. Towards blockchain-based auditable storage and sharing of IoT data. In: *CCSW' 17. Proceedings of the 2017 on Cloud Computing Security Workshop*. 2017:45-50. https://doi.org/10.1145/3140649.3140656
11. Winkler T, Rinner B. Security and privacy protection in visual sensor networks: a survey. *ACM Comput Surv*. 2014;47:2:1-2:42.
12. Chen D, Zhao Q, Chen F, Luo S. Adaptive residual current circuit breaker based on microcontroller. In: *ICDMA 17'. 2011 Second International Conference on Digital Manufacturing & Automation;* 2011: 159-162. https://doi.org/10.1109/ICDMA.2011.46
13. Chen D, Chang G, Jin L, Ren X, Li J, Li F. A novel secure architecture for the internet of things. In: *2011 Fifth International Conference on Genetic and Evolutionary Computing*. IEEE; 2011:311-314. https://doi.org/10.1109/ICGEC.2011.77
14. Abdullah Z, Chen G, Abdullah MA, Chambers JA. Enhanced secrecy performance of multihop IoT networks with cooperative hybrid-duplex jamming. *IEEE Trans Inf Forensics Secur*. 2020;16:161-172.
15. Zhao J. On resilience and connectivity of secure wireless sensor networks under node capture attacks. *IEEE Trans Inf Forensics Secur*. 2017;12(3):557-571. https://doi.org/10.1109/TIFS.2016.2613841
16. Liu D, Liu Z, Li L, Zou X. A low-cost low-power ring oscillator-based truly random number generator for encryption on smart cards. *IEEE Trans Circuits Syst II Express Briefs*. 2016;63(6):608-612.
17. Gu Z, Chen H, Xu P, Li Y, Vucetic B. Physical layer authentication for non-coherent massive SIMO-enabled industrial IoT communications. *IEEE Trans Inf Forensics Secur*. 2020;15:3722-3733.
18. Trappe W, Howard R, Moore RS. Low-energy security: limits and opportunities in the internet of things. *Secur Privacy IEEE*. 2015;13(1):14-21.
19. Angelo F, Luciano A, Andrea P, Antonio P. Using virtual environments for the assessment of cybersecurity issues in IoT scenarios. *Simul Model Pract Theory*. 2017;73:43-54.

20. Ho G, Leung D, Mishra P, Hosseini A, Wagner D. Smart locks: lessons for securing commodity internet of things devices. In: *Proceedings of the 11th ACM on Asia conference on computer and communications security*. 2016:461-472.
21. Chen H, Yu J, Zhou H, Zhou T, Liu F, Cai Z. SmartStore: A blockchain and clustering based intelligent edge storage system with fairness and resilience. *Int J Intell Syst*. 2021;36(9):5184-5209. https://doi.org/10.1002/int.22509
22. Melanie S. *Blockchain: Blueprint for a New Economy*. 1005 Gravenstein Highway North, Sebastopol, CA 95472. O'Reilly Media, Inc.; 2015.
23. Konstantinos C, Michael D. Blockchains and smart contracts for the internet of things. *IEEE Access*. 2016;4: 2292-2303.
24. Chen H, Zhou H, Yu J, et al. Trusted audit with untrusted auditors: a decentralized data integrity crowdauditing model based on blockchain. *Int J Intell Syst*. 2021. https://doi.org/10.1002/int.22548
25. Shawn W, Tome B, Josh B, Vitalik B. Storj Whitepaper V3; 2016. https://www.storj.io/whitepaper
26. Golann GG, Ittai A, Shelly G, Dahlia M, Alin T. SBFT: a scalable and decentralized trust infrastructure. In: 2019 *49th Annual IEEE/IFIP international conference on dependable systems and networks (DSN)*. IEEE; 2019:568-580.
27. Wang G, Shi ZJ, Nixon M, Han S. SMChain: A scalable blockchain protocol for secure metering systems in distributed industrial plants. IoTDI '19 ACM; New York, NY, USA; 2019:249-254.
28. Liu J, Li W, Karame GO, Asokan N. Scalable byzantine consensus via hardware-assisted secret sharing. *IEEE Trans Comput*. 2019;68(1):139-151.
29. MuhammadSalek A, Koustabh D, Fabio A. IoT data privacy via blockchains and IPFS. In: *Proceedings of the seventh international conference on the internet of things*. 2017:1-7.
30. Dorri A, Kanhere SS, Jurdak R, Gauravaram P. Blockchain for IoT security and privacy: the case study of a smart home. In: 2017 *IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*. IEEE. 2017:618-623.
31. Misra S, Mukherjee A, Roy A, Saurabh N, Rahulamathavan Y, Rajarajan M. Blockchain at the edge: performance of resource-constrained IoT networks. *IEEE Trans Parall Distr Syst*. 2021;32(1):174-183.
32. Xu C, Wang K, Li P, et al. Making big data open in edges: a resource-efficient blockchain-based approach. *IEEE Trans Parall Distr Syst*. 2019;30(4):870-882.
33. Wang G, Shi Z, Nixon M, Han S. ChainSplitter: Towards blockchain-based industrial IoT architecture for supporting hierarchical storage. IEEE. 2019:166-175.
34. Benet J. IPFS—Content Addressed, Versioned, P2P File System. *CoRR*. abs/1407.3561; 2014.
35. Yu J, Chen H, Wu K, Zhou T, Cai Z, Liu F. Centipede: Leveraging the distributed camera crowd for cooperative video data storage. *IEEE Internet Things J*. 2021. http://doi.org/10.1109/jiot.2021.3074823
36. Tian Z, Li M, Qiu M, Sun Y, Su S. Block-DEF: A secure digital evidence framework using blockchain. *Inf Sci*. 2019;491:151-165.
37. Karp RM. *Reducibility Among Combinatorial Problems*. Boston, MA: Springer US; 1972: 85-103.
38. Miguel C, Barbara L. Practical byzantine fault tolerance. In: *Proceedings of the Third Symposium on Operating Systems Design and Implementation*. USENIX Association; 1999:173-186.
39. Atapattu S, Ross N, Jing Y, He Y, Evans JS. Physical-layer security in full-duplex multi-hop multi-user wireless network with relay selection. *IEEE Trans Wire Commun*. 2019;18(2):1216-1232.
40. Zhong Y, Han T, Li Q, Ge X. Delay and physical layer security tradeoff in large wireless networks. In: 2018 *IEEE International Conference on Communications (ICC)*. IEEE. 2018:1-7.
41. Diego O, John O. In search of an understandable consensus algorithm. In: 2014 *USENIX Annual Technical Conference (USENIX ATC 14)*. 2014:305-319.

How to cite this article: Yu J, Chen H, Wu K, Zhou T, Cai Z, Liu F. EviChain: A scalable blockchain for accountable intelligent surveillance systems. *Int J Intell Syst*. 2022;37:1454-1478. https://doi.org/10.1002/int.22676