Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

LEChain: A blockchain-based lawful evidence management scheme for digital forensics

Meng Li [a], Chhagan Lal [b], Mauro Conti [c], Donghui Hu [a,*]

[a] Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education, School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, 230601, China
[b] Simula Research Laboratory, Norway
[c] Department of Mathematics, University of Padua, 35131 Padua, Italy

ARTICLE INFO

Article history:
Received 26 November 2019
Received in revised form 19 August 2020
Accepted 28 September 2020
Available online 30 September 2020

Keywords:
Digital forensics
Lawful evidence
Chain of custody Blockchain
Security
Privacy

ABSTRACT

Lawful evidence management in digital forensics is of paramount importance in police investigations because such evidence is used to convict suspects of crimes. Existing studies have adopted cloud computing to collect evidence and then leveraged blockchain to support the transparency, immutability, and auditability of the evidence. Unfortunately, such studies only rely on a weak security model and do not cover the entire life cycle of the evidence or address the key privacy issues, i.e., witness privacy in evidence collection and juror privacy in court trials. In this work, we propose LEChain, a blockchain-based lawful evidence management scheme to supervise the entire evidence flow and all of the court data (e.g., votes and trial results), extending from evidence collection and access during the police investigation to jury voting in the court trials. Specifically, we utilize short randomizable signatures to anonymously authenticate witnesses' identities to protect the witness privacy. Then, we leverage fine-grained access control based on ciphertext-policy attribute-based encryption for evidence access. Next, we design a secure voting method to protect juror privacy. In addition, we build a consortium blockchain to record evidence transactions. Finally, we formally analyze the security and privacy of LEChain and evaluate its computational costs and communication overhead by implementing a prototype based on a local Ethereum test network.

1. Introduction

Digital forensics [1] is a legal procedure to collect, analyze, store, and report digital evidence [2,3]. Such evidence is essential to police investigations because it illustrates the facts of cyber-crime scenes and connects suspects to criminal activities. Therefore, it is crucial to handle the gathered evidence carefully to ensure that it is admissible to police investigations and court trials [4] and to prevent tampering or misconduct. In traditional forensics, the memory limitations of collection devices [5] cause data containing evidence to be continually overwritten over time. While law enforcement can confiscate devices, it is time-consuming and laborious to identify all the pertinent devices located at different and remote locations [6]. All of these scenarios result in delays in reporting evidence. However, it is urgent to report data containing evidence to an investigative cloud to protect lawful evidence and improve the efficiency of investigations. Here, the investigative cloud is a special cloud server that is used for storing evidence for investigations.

To support transparency, immutability, and auditability of when managing such lawful evidence, existing research [2,4,6–9] have resorted to blockchain technology [10–14]. Unfortunately, they only rely on a weak security model; in addition, they do not cover the entire life cycle of the evidence, nor do they address key privacy issues, i.e., witness privacy and juror privacy. In this work, we build a concrete blockchain-based framework to manage the entire flow of evidence and court data (e.g., the charge, plead, votes, and trial results). Our framework extends from evidence collection and access during the police investigation to the jury voting in the court trial. The life cycle of the evidence is depicted in Fig. 1 which is an extended version of the digital forensics progress model in block4forensics [2]. It consists of collection, examination, analysis, reporting, the court trial, and settlement. Police investigators collect evidence from crime scenes and witnesses. After examination, the police investigators store and upload the evidence for a further analysis conducted by a crime scene analyst. When the analysis is completed, the analyst reports the analysis result. During the court trial, evidence is further accessed and jurors will vote on whether the defendant is guilty or not. Finally, the judge awards a settlement. In the analysis phase, we follow the idea of block4forensics [2], which incorporates different types of data to conduct evidence analyses.

* Corresponding author.
E-mail addresses: mengli@hfut.edu.cn (M. Li), chhagan@simula.no (C. Lal), conti@math.unipd.it (M. Conti), hudh@hfut.edu.cn (D. Hu).

https://doi.org/10.1016/j.future.2020.09.038
0167-739X/© 2020 Elsevier B.V. All rights reserved.

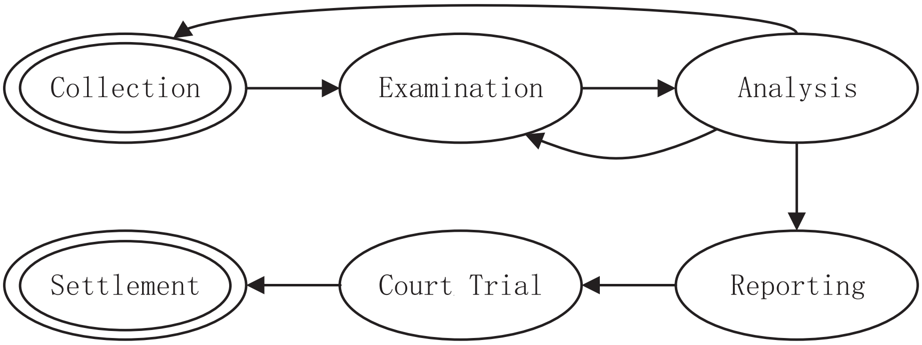


Fig. 1. life cycle of evidence.

The research questions and problems in this work focus on three aspects of digital forensics. First, we focus on how to manage the evidence throughout its entire "lifetime" in a verifiable and tamper-proof manner, including authorizing entities to upload evidence, authenticating entities that access evidence, and revoking the upload/access rights of targeted entities. For instance, only some entities (e.g., witnesses, police investigators, crime scene analysts, and monitoring devices) can submit their collected evidence to the blockchain. Then, only the entities with a specific attribute set can access the evidence via the blockchain. If an entity has gone rogue or has been corrupted, we need to be able to revoke his/her attributes to disqualify the entity in the blockchain network. Such malicious entities could engage in hostile activities. For example, Bhadresh Gohil, an ex-solicitor, has been accused of forging documents to frame Scotland Yard detectives [15].

Second, we emphasize how to protect privacy during evidence management, namely witness privacy [1,8,16] in the chain of custody and juror privacy in court trials. Witnesses fear intentional retaliation [17] when they provide evidence at a crime scene or in a police department. Therefore, it is vital to protect their identities against potentially malicious entities, e.g., defendants and their accomplices. Otherwise, they may choose not to provide valuable information in an investigation, which may then be suspended. Jurors vote on whether a defendant is guilty or not, and a judge sentences the defendant according to the votes of the jury. Consequently, considerable amounts of efforts are investigated in determining jurors' names, home addresses, and votes by people who try to manipulate the trial results. Therefore, the jurors' identities and votes need to be kept a secret to guarantee their safety [18]. Meanwhile, we need to ensure the verifiability of the correctness of the trial results given that each juror's vote is not directly presented in plaintext.

Third, we consider how to defend against internal adversaries, e.g., honest-but-curious crime scene analysts and any corrupt police investigators. For the honest-but-curious entities, our objective is to protect privacy as mentioned earlier. However, in addition, there may be corrupt individuals, for example, two corrupt detectives were jailed for sabotaging child abuse by forging documents and concealing evidence [19]. Violations in proper evidence collection can have dramatic consequences, necessitating validation of the evidence throughout the chain of custody. Finally, the computational cost and communication overhead need to be kept within an acceptable range for the blockchain-based system to run sustainably.

To solve these problems, we propose LEChain, which is a blockchain-based lawful evidence management scheme. The LEChain scheme consists of six phases: system initialization, entity registration, evidence collection, evidence upload, evidence access, and court trial and sentences. A trusted authority initializes the whole system, and entities register to this trusted authority. Victims, witnesses, and monitoring devices provide evidence to a police investigator. Crime scene analysts analyze the evidence and send their analysis results to the police investigator, who uploads hash values of all the evidence to the blockchain and stores all the evidence in the police department. Entities can access the evidence via the blockchain if they possess a specific attribute set. Lawyers issue a charge or plead for the accused during the court trial and upload corresponding data to the blockchain. Each juror casts a vote to the blockchain. The judge computes the voting result and uploads the hash value of the court data to the blockchain. In addition to the above operations, there are subsequent operations related to other parties that are also involved in the life-cycle of digital forensics processing. These entities include prisons and insurance companies. We further explain these entities and their possible roles in Section 3.1.

In this study, we make the following contributions:

- We propose the design of LEChain, a blockchain-based lawful evidence management scheme, to better control the entire chain of evidence in digital forensics with transparency, unforgeability, and verifiability. Specifically, we utilize short randomizable signatures to authenticate witnesses' identities anonymously. We leverage fine-grained access control based on ciphertext-policy attribute-based encryption (CP-ABE) [20] to upload, access, and update evidence. We design a secure voting method [21] to protect juror privacy, and we build a consortium blockchain to store all the evidence transactions to provide a transparent, immutable, and auditable supervision of the evidence.
- To the best of our knowledge, we are the first to address two key privacy issues, i.e., witness privacy in evidence collection and juror privacy in court trials in forensics investigation. LEChain aims to protect the identities of the witnesses and jurors, as well as the votes of the jurors. We establish a stronger security model for digital forensics. In particular, we adopt the honest-but-curious security assumption for most entities, but we include the possibilities of corrupt police investigators in the security model; such entities can launch several attacks, namely data injection, data falsification, and unauthorized access. Finally, we prove the security and privacy properties of the LEChain scheme.
- To show the deployment feasibility and efficiency of LEChain, we build a prototype based on Ethereum[1] test network and proof-of-authority (PoA)[2] consensus mechanism to evaluate its performance concerning the computational cost, communication overhead, and network latency. We also compare LEChain to existing works with respect to computational cost and communication overhead.

[1] https://www.ethereum.org

[2] https://www.poa.network

The remaining of this paper is organized as follows. We discuss related work in Section 2. In Section 3, we present the system model, security model, and design goals. Some preliminaries are reviewed in Section 4. We present the design and working methodology of the proposed LEChain scheme in Section 5, followed by security analysis and performance evaluation in Section 6 and Section 7, respectively. We discuss some related issues in Section 8. Finally, we conclude our work in Section 9.

2. Related work

In literature, there exist few research works that proposes blockchain-based digital forensics solutions to improve one or more aspects of digital forensics domain. We introduce some related works from three aspects: digital forensics, privacy in digital forensics, and blockchain technology for digital forensics.

2.1. Digital forensics

A typical process of digital forensics consists of four phases: collection, examination, analysis, and reporting [2,22]. Specifically, collection is the process of a law-executor's collecting evidences aiming to gather investigation related data from entities or devices. Then the collected data are stored in a protected department. Examination performs search and access of the related data and verify the data. Analysis is the process of an analyst's conducting professional (physical or chemical) analysis on the collected data to further provide information useful to the investigation. Reporting is to provide a final investigation report.

2.2. Privacy in digital forensics

Nieto et al. [8] focused on the digital witness [1], conducted a detailed analysis of the digital witness approach with regards to privacy, and proposed digital anonymous witnessing to bridge forensic computing with witnesses. They claimed that privacy issues would arise if some information (which are not closely related to the investigation) could be deduced or unauthorized entities could infer the information. Meanwhile, they mentioned that privacy in digital witness included anonymity, unlinkability and location privacy.

Frankle et al. [23] pointed out that the US federal court system is marching to guarantee the accountability of electronic surveillance. They presented a framework for addressing the accountability problem as a secret information process by using cryptographic commitments, zero-knowledge proofs, secure multiparty computation, and secret sharing. Specifically, a judge issues an order and puts a commitment and metadata to the order. An agency leverages the order to access data from a data source and puts a commitment and a zero-knowledge proof to this request. If the data source replies with corresponding data, it puts a commitment to its response. In the end, the system can guarantee that confidentiality is perfect (information theoretic). However, there are more than one judges in their system model and they did not include the court trial.

2.3. Blockchain technology for digital forensics

Zhang et al. [7] claimed that there was no available method for improving the trustworthiness of the interaction records of entities in cloud forensics. Here, cloud forensics refers to combining cloud computing and digital forensics to focus on the collection of digital evidence from a cloud infrastructure. To address it, they proposed a proof of existence and privacy-preserving scheme PPCF for process records using blockchain and group signatures. Their focus is only on transferring forensics data, which lacks other essential operations in digital forensics, such as fine-grained data access.

Cebe et al. [2] stated that connected and smart vehicles would provide valuable data to different stakeholders, such as maintenance companies, vehicle manufacturers, drivers, and insurance companies, which had an important effect on vehicular forensics. The authors presented a permissioned blockchain-based framework for managing vehicular data, which combines the vehicular public key infrastructure with the blockchain to establish membership and protect privacy. Later they proposed a fragmented ledger to preserve detailed vehicular data, e.g., diagnosis records and maintenance reports. However, this framework is only applicable to vehicular forensics, and they do not address fine-grained data access.

Le et al. [6] mentioned the identity privacy in IoT forensics. They presented a permissioned blockchain-based framework called BIFF to enhance the integrity and traceability of the gathered IoT evidence. To mitigate identity privacy concern, they integrate a digital certificate scheme into the Merkle signature. Lone et al. [4] proposed a blockchain-based digital forensics chain of custody to provide integrity and tamper resistance of digital evidence. Their process model includes four functions: evidence creation, evidence transfer, evidence deletion, and evidence display. Unfortunately, they ignored the possibility of malicious police investigators and neglected privacy protection. Additionally, the evidence deletion was a disaster for police investigators, which should not be included in the functions.

Tian et al. [9] proposed a digital evidence framework called Block-DEF, which is based on blockchain to support evidence collection, storing, verifying, and retrieving. Their design stores the evidence information on the blockchain and preserves the evidence on a trusted storage platform. They used two multi-signature schemes for evidence submission and retrieval to guarantee traceability. However, their evidence access phase does not consider the attributes of evidence requesters. Further, it suffers from high communication overhead incurred by redundant interactions among an evidence provider, a content provider, an evidence requester, and the trusted storage.

3. Problem statement

In this section, we define the system model, security model, and design goals for our proposed scheme.

3.1. System model

The system model of LEChain consists of a trusted authority, a police department, a court, a prison, victims, prosecution lawyers, defense lawyers, police investigators, crime scene analysts, witnesses, monitoring devices, a judge, a jury (with jurors), and an insurance company as depicted in Fig. 2. The key notations are explained in Table 1.

- Victim: provides allegations and evidence to a police investigator.
- Witness: is a crucial information source of crime scenes. It provides evidence for a crime (e.g., a description of what happened to a victim) to a police investigator.
- Monitoring Device: is deployed by different organizations in diverse locations. A police investigator extracts the recorded video surveillance from these devices.
- Police Investigator: gathers evidence, uploads a hash value of evidence to the blockchain, and it stores all the evidence in the police department. Then the evidence could be verified by victims, witnesses, monitoring devices, and crime scene analysts. If there is a mismatch, the original

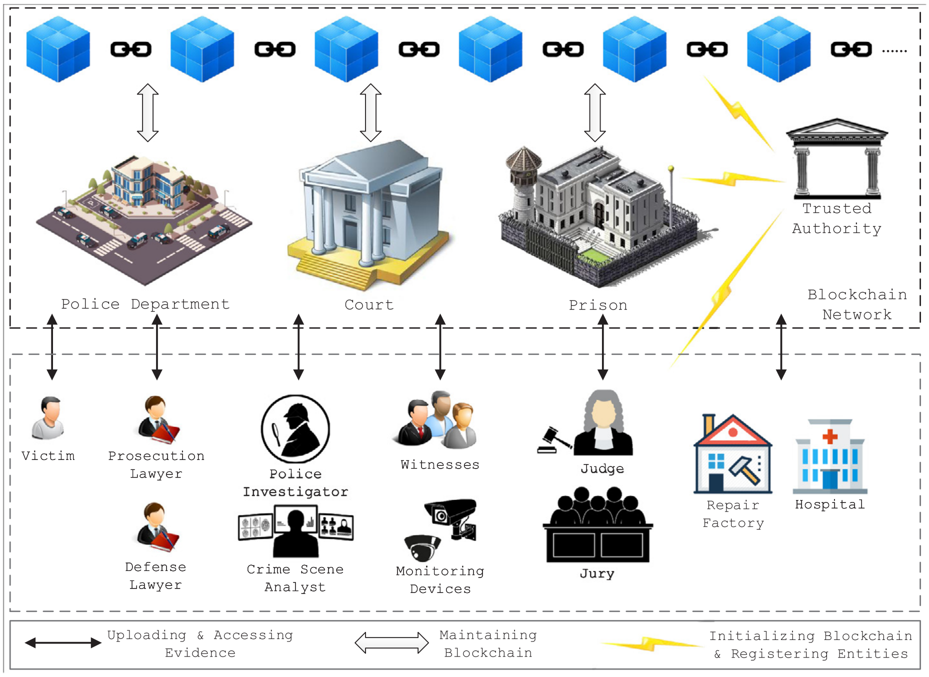


Fig. 2. System model of LEChain.

data provider can report to the trusted authority about this inconsistency and prove it by showing the original data as well as the signature of the police investigator. The evidence could be descriptions in audio and video footage [24].

- Crime Scene Analyst: analyzes evidence, sends analysis results, or newly discovered evidence to a police investigator. The crime scene analyst can also access evidence by sending an access transaction to the blockchain network and then receiving the evidence from a data provider (e.g., police department storage).
- Prosecution Lawyer: issues a charge during court trial and uploads corresponding data to the blockchain.
- Defense Lawyer: pleads during the court trial and uploads corresponding data to the blockchain.
- Juror: generates a vote and casts it in the blockchain in a privacy-preserving and verifiable way. An original vote is either 0 (not guilty) or 1 (guilty).
- Judge: computes and verifies the voting result, generates a trial result (i.e., judgment, recess, or retrial), and submits the hash value of court data to the blockchain.
- Police Department: stores all the evidence, and tallies votes. Here, "tally votes" means that the vote collector, i.e., three authorities in LEChain, processes the slices of the votes from the jurors and obtain a partial voting result.
- Court: holds court trials, preserves court data, and tallies votes. Transparency in the court proceedings refers to the transparency of different types of documents, including the court rules, court agenda, case files, resolved and pending, etc. When carried out properly, it leads to increased efficiency of police investigation as well as promotes public confidence in the legal system [25].
- Prison: detains prisoners, keeps prisoners' records, and tallies votes.
- Trusted Authority (*TA*): is a governmental agency, and it initializes the whole system. The motivation to use *TA* has two aspects. First, we believe there is a governmental agency that is not easily compromised given hardware-protected protocol running environments [26,27], rigorous monitoring, and detailed access logs. Hence, it can perform as a trusted authority. Second, the *TA* only works in system initialization, entity registration, and entity tracking. The first two phases do not conflict with the blockchain design. The tracking function is considered here because different from Bitcoin system or other blockchain-based financial systems, the evidence management requires the ability to locate malicious insiders and protect justice. Based on the second reason, if the participants generate cryptographic key material themselves, it would be more difficult to reveal their real identities.

Please note that in LEChain, the TA, the police department, the court, and the prison are the four blockchain miners. The miners co-maintain the blockchain by validating upload/access transactions and running PoA as their consensus mechanism. All the above entities first register to the *TA*. Any entities that need access to a piece of evidence has to go via blockchain. However, to get access to the blockchain, the entity has to be entitled to some predefined set of attributes. In particular, the *TA* provides a set of attributes to an entity according to its roles in different police investigations.

3.2. *Security model*

During digital forensics investigation and usage, various security and privacy threats could arise from internal or external adversaries. Most internal entities will strictly follow the protocols and faithfully submit and access evidence or court data, but they may try to learn the identity of witnesses and jurors. The privacy is protected against a small part of malicious entities, i.e., police investigators, which can launch several attacks, namely data injection, data falsification, and unauthorized access.

Specifically, data injection means that malicious entities produce some new evidence which is not consistent with the facts or obtained from any victim, witness, or monitoring device. Then the entities attempt to upload the new evidence to the blockchain. Data falsification indicates that malicious entities, especially the malicious police investigators and crime scene analysts, try to tamper with the collected evidence and upload the forged evidence to the blockchain. The judge may deviate from the result

Table 1
Key notations.

Notation	Definition
p; $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$	Prime number; Cyclic group
$g, \hat{g}$; e	Group generator; Bilinear pairing
$\mathcal{H}$	Secure hash function
$(x, y), (g, X, Y)$	Group secret key, Group public key
N	Number of attributes
avk, pak	Attribute version key, Public attribute key
vt, w, pi	Victim, witness, Police investigator
sa, jr, jg	Crime scene analyst, Juror, Juror
pl, dl	Prosecution lawyer, Defense lawyer
pd, co, pr	Police department, Court, Prison
s_w, pk_w, sc_w	Secret key, Public key, Secret of w
$gsk_w, (u_w, \hat{\sigma}_w, e(\hat{\sigma}_{w1}, Y))$	Group secret key of w
$usk_{sa}, (S, L, S_x)$	User secret key of sa
E_{vt}, E_w	Evidence of vt, Evidence of w
$c_w, (c_w^2, c_w^2)$	Ciphertext of w to pi
$R_w, (c_w, \hat{\sigma}'_{w1}, \hat{\sigma}'_{w2}, ch_w, ss)$	Witness report
$Tx_{pi}^{up}, Tx_{sa}^{ac}$	Upload transaction, Access transaction

of the jurors. The misconduct of a judge is possible. A real world scenario is that a misbehavior by a handful of judges across the United States was reported by Cynthia Gray, the director of the American Judicature Society Center for Judicial Ethics at the time. Among them, a New York judge tried to intercede with other judges to help her friends in two child custody cases [28].

Unauthorized access refers to malicious entities' attempting to access some evidence via the blockchain, and these entities do not hold the pertinent attributes. Collusion attacks among the police department, court, and prison are not considered. However, external adversaries can eavesdrop on the communication channel, launch an impersonation attack and replay attack, in an attempt to violate privacy, and sabotage the system. For the sake of simplicity, by malicious entities, we refer to a small number of police investigators, crime scene analysts, and lawyers.

3.3. Design goals

The key design goals that LEChain aims to achieve are as follows.

- **Privacy**: LEChain protects the privacy of the following entities in the system.
  - *witness privacy:* when a witness provides evidence to a police investigator, the police investigator uploads a hash value of evidence to the blockchain and preserves the evidence in the police department. In this process, the other entities must not be able to identify the witness' identity or link one witness' uploading behaviors at different times.
  - *juror privacy:* when a juror casts a vote to the blockchain, the other entities cannot identify the juror's identity or acquire the real vote.
  - *data privacy*: refers to the restriction that a police investigator cannot obtain the evidence that is beyond her/his access qualification when interacting with the blockchain.
- **Authentication**: the identity of an entity which uploads (accesses) to (via) the blockchain should be authenticated to prevent illegal entities from upload (access) blockchain data.
- **Access Control**: the attributes of an entity which accesses the blockchain should be verified whether they are eligible to prevent unauthorized entities from accessing blockchain data.
- **Integrity & Auditability**: the system should guarantee the integrity and auditability of blockchain data such that the data are hard to tamper with and easy to be audited.
- **Traceability**: the source of evidence and court data should be able to be tracked. We will further discuss it in detail in Section 6.5.
- **Efficiency**: LEChain aims to provide efficiency concerning (i) computational cost, i.e., use of a lightweight process for uploading and accessing evidence in blockchain, and (ii) communication overhead, i.e., the total length of a blockchain data transaction should be as short as possible so as to save network bandwidth.

4. Preliminaries

4.1. Anonymous authentication

The short randomizable signature scheme [29] mainly achieves anonymous authentication of users' identities. It consists of the following functions:

- $\mathsf{Setup}_1(\lambda)$: given a security parameter λ, outputs public parameters pp_1.
- $\mathsf{Keygen}(pp_1)$: given public parameters pp_1, selects $\hat{g}$ and a signing key $sk = (x, y)$, and calculates a corresponding verification key $pk = (\hat{g}, \hat{X}, \hat{Y})$.
- $\mathsf{Sign}(sk, m)$: given a signing key sk and a message m, selects a random number r and outputs a signature σ.
- $\mathsf{Verify}(pk, m, \sigma)$: given a verification key pk, a message m and a signature σ, verifies the σ. If the verification passes, outputs 1; and 0 otherwise.
- $\mathsf{GSetup}(\lambda)$: a group manager executes Setup and Keygen to obtain $sk = (x, y)$ and $pk = (\hat{g}, \hat{X}, \hat{Y})$, sets a group public key gpk as pk with a generator g, set a group secret key gsk as sk, and a selects a hash function $\mathcal{H}$.
- $\mathsf{PKIJoin}(i, \lambda)$: a user i generates $(sk_i, pk_i) \leftarrow \mathsf{Keygen}(pp_1)$ and sends pk_i to a certificate authority.
- GJoin: user i generates a secret s_i and sends $(\tau, \hat{\tau}) \leftarrow (g^{s_i}, \hat{Y}^{s_i})$ with a signature η to the group manager, which

validates η and $(\tau, \hat{\tau})$. If user i passes a Schnorr's interactive proof of knowledge [30] of s_i, the group manager chooses a random number h and calculates $\hat{\sigma} \leftarrow (\hat{\sigma}_1, \hat{\sigma}_2)$. Lastly, the group manager records $(i, \tau, \eta, \hat{\tau})$ and sends $\hat{\sigma}$ to user $i(s_i, \hat{\sigma}, e(\hat{\sigma}_1, \hat{Y}))$ as the gsk_i.

- $\mathsf{GSign}(gsk_i, m)$: user i randomizes σ by choosing a random number u and calculating $(\hat{\sigma}'_1, \hat{\sigma}'_2) \leftarrow (\hat{\sigma}^u_1, \hat{\sigma}^u_2)$, chooses a random number t and calculates $e(\hat{\sigma}_1, \hat{Y})^u \leftarrow e(\sigma_1, \hat{Y})^{tu}$, $c \leftarrow \mathcal{H}(\hat{\sigma}'_1, \hat{\sigma}'_2, e(\hat{\sigma}_1, \hat{Y})^{t \cdot u}), m)$, calculates $ss \leftarrow u + c \cdot s_i$ and outputs a group signature μ_i on m.
- $\mathsf{GVerify}(gpk_i, m, \mu_i)$: a verifier verifies a signature μ_1 on message m, and outputs 1 if the verification passes and 0 otherwise.
- $\mathsf{GOpen}(gsk, m, \mu_i)$: the group manager looks up all the items $(i, \tau_i, \eta_i, \hat{\tau}_i)$ and returns a matched (i, τ_i, η_i) along with a proof of knowledge of a valid $\hat{\tau}_i$.

4.2. Attribute-based access control

The attribute-based access control scheme [20] guarantees that evidence access operations are legal and the access rights of traitors are revoked in time. It consists of the following functions:

- $\mathsf{Setup}_2(\lambda)$: given a security parameter λ, outputs a master key msk, public parameters pp_2 and a set of public attribute keys $\{pak_x\}$.
- $\mathsf{USKeyGen}(msk, A, \{vk_x\}_{x \in A})$: given the master key msk, a set of attributes A, and a set of attribute version keys $\{avk_x\}_{x \in A}$, outputs a user secret key usk.
- $\mathsf{Encrypt}(pp_2, \{pak_x\}, m, \mathcal{S})$: given the public parameters pp_2, a set of public attribute key $\{pak_x\}$, a message m and an access structure $\mathcal{S}$, encrypts m to output a ciphertext C. Only users with a set of attributes satisfying $\mathcal{S}$ can decrypt C.
- $\mathsf{Decrypt}(C, usk)$ given a ciphertext C including an access structure S and a user secret key usk for a set of attributes A. If A satisfies S, decrypts C to return a message m.
- $\mathsf{UKeyGen}(msk, vk_{\hat{x}})$: given the master key msk and the current version key $vk_{\hat{x}}$ of the revoked attribute $\hat{x}$, outputs a new version key $\widehat{vk}_{\hat{x}}$ of $\hat{x}$ and an update key $uk_{\hat{x}}$.
- $\mathsf{USKUpdate}(usk, uk_{\hat{x}})$: given the current user secret key usk and an update key $uk_{\hat{x}}$ of the revoked attribute $\hat{x}$, outputs a new user secret key $\widehat{usk}$.
- $\mathsf{CUpdate}(C, uk_{\hat{x}})$: given a ciphertext C and an update key $uk_{\hat{x}}$, outputs a new ciphertext $\widehat{C}$.

4.3. Blockchain

Initially serving as a fundamental technique in Bitcoin [10, 14], a blockchain BC is a public ledger which links a continuously growing chain of blocks with cryptographic hashes. A block contains a certain number of transactions between users, and each block is created by a selected miner based on a consensus mechanism. A miner can receive some rewards as participation incentives.

A consortium blockchain [12] CBC is a permissioned blockchain maintained by a group of identified parties, and it secures transactions between users who do not fully trust each other but share a common goal. Such CBC verifies all internal transactions conducted among users. The users select a winning user according to group consensus in each period. The elected user creates a new data block. The new block is chained to the last block on the blockchain. CBC is already adopted in several applications, such as vehicular networks [13] and searchable encryption [31].

5. The proposed scheme LEChain

In this section, we describe the design and working methodology of our proposed LEChain scheme by presenting an overview and six detailed phases.

5.1. Overview

The *TA* initializes the whole management system by generating public parameters and cryptographic keys. It cooperates with the police department, the court, and the prison in building a consortium blockchain for storing all the upload transactions, access transactions, and other pertinent transactions. Entities, such as witness, police investigator and crime scene analyst, register to the *TA* and obtain keys when joining in the system.

A witness and a police investigator co-upload new evidence to the blockchain via using anonymous authentication and signatures and sending an upload transaction. Registered entities of a specific attribute set access the evidence by sending an access transaction to the blockchain network. For example, a crime scene analyst queries the blockchain to access some video footage previously uploaded by a police investigator. The blockchain nodes first validate her/his identity in the access transaction. If the validation passes, they will search for pertinent evidence by hash values and return a download link to the analyst. The analyst retrieves the encrypted footage via private information retrieval from the distributed storage systems and decrypts it by recovering encryption keys. LEChain suggests the use of off-chain[3] storage of evidence in blockchain as it provides high throughput, high capacity, low latency, permissioning, and rich query support.

Lawyers issue a charge or pleads during court trial(s) and upload corresponding data to the blockchain. Each juror generates a vote and casts it to the blockchain by using slicing in a privacy-preserving and verifiable way. The three authorities tally votes. The judge verifies the vote result and generates a trial result, preserves court data in the court, and uploads a hash value of court data to the blockchain.

Next, we present the details of the main framework of LEChain, which consists of six phases, namely, system initialization, entity registration, evidence collection, evidence upload, evidence access, court trial, and subsequences.

5.2. System initialization

The *TA* initializes the whole system by generating public parameters and cryptographic keys. First, the *TA* chooses a security parameter λ, generates public parameters $pp_1 \leftarrow (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \mathcal{H})$, sets $sk = (x, y)$ and $pk = (\hat{g}, \hat{X}, \hat{Y})$ where $\hat{g} \overset{\$}{\leftarrow} \mathbb{G}_2$ and $(\hat{X}, \hat{Y}) \leftarrow (\hat{g}^x, \hat{g}^y)$. The *TA* sets a group public key gpk as pk with a generator $g \in G_1$ and a secret key gsk as sk. $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ are three cyclic groups of prime order p and e is a bilinear map: $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. $\mathcal{H}$ is a secure hash function.

Next, the *TA* chooses four random numbers $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{Z}_p$ as the master key msk and generates public parameters $pp_2 = (g, g^{\alpha_1}, g^{1/\alpha_2}, g^{\alpha_2}, e(g, g)^{\alpha_4})$. After communicating with the three miners (i.e., police department, court, and prison), the *TA* receives a universe of attributes $\mathcal{UA} = \{x_i\}^l_{i=1}$. For each attribute x, the *TA* randomly chooses a number $v_x \in \mathbb{Z}_p$ as the initial attribute version key $avk_x = v_x$ and computes a public attribute key $pak_x = (pak^1_x, pak^2_x) = (\mathcal{H}(x)^{v_x}, \mathcal{H}(x)^{v_x \alpha_3}))$.

Last, the *TA* initializes a CBC with the three miners. They divide time to a sequence of slots $\{sl_1, sl_2, \ldots\}$. Each miner M_i sets a local ledger $CBC_i = B_0$, which is the genesis block B_0 including an empty blockheader, miners' identities pd, co, and pr, public keys pk_{pd}, pk_{co} and pk_{pr}, a timestamp, and each miner's signature on previous items.

[3] Off-chain strategy only needs to store a pointer to data which are stored in distributed databases such as InterPlanetary File System (IPFS) [32].

5.3. Entity registration

Entities that wish to upload and access data via the blockchain will first register to the *TA*. For the sake of simplicity, we only use one victim vt providing evidence to a police investigator, one witness w providing evidence, one police investigator *pi* uploading evidence, one crime scene analyst *sa* accessing data from the police department, twelve jurors $\{jr_i\}_{i=1}^{12}$ submitting votes, one prosecution lawyer *pl* issuing a charge, one defense lawyer *dl* pleading, and one judge *jg* uploading court data.

The police department *pd*, court *co*, and prison *pr* register to obtain private keys sk_{pd}, sk_{co}, sk_{pr} and public keys $pk_{pd} = g^{sk_{pd}}$, $pk_{co} = g^{sk_{co}}$, $pk_{pr} = g^{sk_{pr}}$, respectively. Similarly, *pi*, *pl*, *dl*, and *jg* obtain private and public keys.

For anonymous authentication, w chooses a random number $s_w \in \mathbb{Z}_p$ as her secret key and computes $pk_w = g^{s_w}$ as her public key. Then she randomly chooses a secret $sc_w \in \mathbb{Z}_p$ and sends a pair $(\tau_w, \hat{\tau}_w) \leftarrow (g^{sc_w}, \hat{Y}^{sc_w})$ with signature $\eta_w \leftarrow \mathsf{Sign}(s_w, \tau_w)$ to the *TA*. The *TA* validates η_w and $(\tau_w, \hat{\tau}_w)$ by checking whether $e(\tau_w, \hat{Y}) \overset{?}{=} e(g, \hat{\tau}_w)$. If w passes a Schnorr's interactive proof of knowledge of s_w, the *TA* chooses a random number $u_w \in \mathbb{Z}_p$ and calculates

$$\hat{\sigma}_w = (\hat{\sigma}_{w1}, \hat{\sigma}_{w2}) = (g^{u_w}, (g^x \cdot \tau_w^y)^{u_w}). \tag{1}$$

Finally, the *TA* records $(w, \tau_w, \eta_w, \hat{\tau}_w)$ and sends $\hat{\sigma}_w$ to w who sets her group secret key

$$gsk_w = (u_w, \hat{\sigma}_w, e(\hat{\sigma}_{w1}, Y)). \tag{2}$$

Similarly, $jr_i (1 \leq i \leq 12)$ registers and receives $gsk_i = (jr_i, \hat{\sigma}_i, e(\hat{\sigma}_i^1, Y))$.

For access control, *pd* generates an item key *ik*. *sa* first registers to one of the three authorities (i.e., police department, court, and prison) and obtain a set of attributes A and a corresponding signature according to his role. Then *sa* registers to the *TA* which verifies the signature. If it is valid, the *TA* randomly chooses a number $o \in \mathbb{Z}_p$ and generates *sa*'s user secret key as

$$usk_{sa} = (S = g^{\frac{\alpha_1 + \alpha_4 o}{\alpha_2}}, L = g^o, \forall x \in A : S_x = g^{o\alpha_2^2 \cdot \mathcal{H}(x)^{v_x o \alpha_2}}) \tag{3}$$

CA sends usk_{sa} to *sa* via a secure channel.

5.4. Evidence collection

A police investigator *pi* sends a collection request transaction Tx_{pi}^{Req} containing a string "Request", the identity *pi* and a timestamp *ts*. The victim vt has some evidence E_{vt} regarding a suspect or a crime scene, and it sends the evidence to the *pi*. Here, we assume that E_{vt} includes the identity of vt. Then *pi* generates a signature σ_{pi}^1 on E_{vc} and returns it to vt. Next, *pi* stores E_{vt} in the police department and computes a hash value $\mathcal{H}(E_{vt})$. The witness w also has some potentially valuable evidence E_w about the crime scene and prepares a witness report R_w as follows:

- Encrypt E_w by choosing a random number $r_0 \in \mathbb{Z}_p$ and computing a ciphertext $c_w = (c_w^1, c_w^2)$ where $c_w^1 = g^{r_0}$, $c_w^2 = E_w \cdot pk_{pi}^{r_0}$.
- Choose a random number u and calculate $(\hat{\sigma}'_{w1}, \hat{\sigma}'_{w2}) \leftarrow (\hat{\sigma}_{w1}^u, \hat{\sigma}_{w2}^u)$.
- Choose a random number t and calculate $e(\hat{\sigma}_{w1}, \hat{Y})^u \leftarrow e(\sigma_{w1}, \hat{Y})^{tu}$, $ch_w \leftarrow \mathcal{H}(\hat{\sigma}'_{w1}, \hat{\sigma}'_{w2}, e(\hat{\sigma}_{w1}, \hat{Y})^{t \cdot u}, c_w)$, $ss \leftarrow u + ch_w \cdot sc_w$ and output a group signature $\mu_w = (\hat{\sigma}'_{w1}, \hat{\sigma}'_{w2}, ch_w, ss)$ on c_w.
- Form a witness report:

$$R_w = (c_w, \hat{\sigma}'_{w1}, \hat{\sigma}'_{w2}, ch_w, ss). \tag{4}$$

Then, w sends R_w to *pi*.

Please note that a witness or victim in this phase could also upload unlawfully evidence to a police investigator. But we can assess digital evidence admissibility in investigation and court trial, which enables the police investigators, lawyers, and judges to determine the evidential weight of evidence using factor analysis. The process includes three phases: digital evidence assessment, digital evidence consideration, and digital evidence determination. We refer the interested readers to [33] for more details.

5.5. Evidence upload

Upon receiving R_w from a witness w, the police investigator *pi* first verifies R_w as follows:

- Compute $D_w \leftarrow (e(\sigma'^{-1}_{w1}, \hat{X}) \cdot e(\sigma'_{w2}, \hat{g}))^{-ch_w} \cdot e(\sigma'^{ss}_{w1}, \hat{Y})$.
- Check $ch_w \overset{?}{=} \mathcal{H}(\hat{\sigma}'_{w1}, \hat{\sigma}'_{w2}, D_w, c_w)$.

If the verification passes, *pi* generates a signature ρ_{pi}^w on R_w, and returns μ_{pi} to w. Then, *pi* decrypt c_w to obtain E_w and stores E_w in the police department. The signature μ_{pi}^w is a proof such that the witness can prove the authenticity of her evidence in the future. Next, *pi* computes a hash value $\mathcal{H}(c_w)$, attaches a timestamp *ts*, generates a signature ρ_{pi}^{up} on $\mathcal{H}(c_w)$ and *ts*, and uploads an upload transaction Tx_{up} to the blockchain network:

$$Tx_{pi}^{up} = (\text{“Upload"}, pi, R_w, \mathcal{H}(c_w), ts, \rho_{pi}^w).$$

If the ρ_{pi}^w is not valid, the Tx_{up} will be rejected by a blockchain node which sends a rejection transaction. Otherwise, the *pd* stores the newly uploaded evidence and sends a store transaction.

5.6. Evidence access

For entities to access the evidence E_w which is stored in the police department, *pd* processes E_w data as follows.

- Partition E_w into several items as $E_w = (I_1, I_2, \ldots, I_n)$ based on the logic granularities. For instance, a description of a suspect can be divided into {sex, height, color of hair, glass, age}.
- Encrypt each item I_i with different item keys ik_i $(1 \leq i \leq n)$ by using AES encryption.
- Define an access structure $\mathcal{T}$ over the universe of attributes $\mathcal{UA}$ for each item key ik_i $(1 \leq i \leq n)$.
- Encrypt ik_i under $\mathcal{T}$ by executing $\mathsf{Encrypt}$: assume $\mathcal{T}$ is a $l \times n$ matrix where l is the number of attributes; randomly select an encryption exponent $z \in \mathbb{Z}_p$ and a vector $\overrightarrow{v} = (z, f_2, \ldots, f_n)$ where $f_2, \ldots, f_n$ share z; for $i = 1$ to l, calculate $\eta_i = \overrightarrow{v} \times \mathcal{T}_i$, where $\mathcal{T}_i$ is a vector of the ith row of $\mathcal{T}$; choose random numbers $b_1, b_2, \ldots, b_l \in \mathbb{Z}_p$ and compute a ciphertext of E_w:

$$C_{pd}^w = (C_1, C_2, C_{3i}, C_{4i}, C_{5i}), \tag{5}$$

where $C_1 = ike(g, g)^{\alpha_1 z}$, $C_2 = g^{\alpha_2 z}$, $C_{3i} = g^{\alpha_4 \eta_i}(g^{\alpha_2})^{-b_i} \mathcal{H}(\delta(i))^{-b_i v_{\delta(i)}}$, $C_{4i} = \mathcal{H}(\delta(i))^{v_{\delta(i)} b_i \alpha_3}$, $C_{5i} = g^{b_i/\alpha_2}(1 \leq i \leq l)$, and function δ associates rows of $\mathcal{T}$ to attributes.

The crime scene analyst *sa* with a set of attributes A can access the needed evidence E_w by reconstructing the encryption exponent z (firstly appeared in Section 5.6), sending an access transaction Tx_{sa}^{ac} to the blockchain network, and receiving a download link. Given the link, private information retrieval (PIR) technique can be adopted when retrieving the data from the distributed storage systems [34]. After that, *sa* decrypts the received data as follows:

- *sa* defines $F \subset \{1, 2, \ldots, l\}$ as $F = \{i : \delta(i) \in A\}$, chooses a set of constants $\{a_i \in \mathbb{Z}_p\}_{i \in F}$, reconstructs $z' = \sum_{i \in F} a_i \eta_i$ if $\{\eta_i\}$ are valid shares of z based on $\mathcal{T}$, encrypts z' using the public keys of three authorities to obtain a ciphertext set $\{c_{sa}^i\}$, attaches a timestamp ts_{sa}, generates a signature ρ_{sa}^{pd} on $\{c_{sa}^i\}$ and ts_{sa}, and forms an access transaction:

$$Tx_{sa}^{ac} = (\text{“Access''}, sa, \{c_{sa}^i\}, ts_{sa}, \rho_{sa}). \tag{6}$$

- The blockchain miners validate identities and attributes in the transaction by verifying ρ_{sa} and decrypted z' from $\{c_{sa}\}$. If the validation passes, they search for pertinent evidence by hash values and return it to *sa*. In this case, *pd* sends C_{pd}^w to *sa*.
- After receiving C_{pd}^w, *sa* calculates:

$$\frac{e(C_2, S)}{\Pi_{i \in F}(e(C_{3i}, L)e(C_{5i}, S_{\delta(i)}))^{a_i}} = e(g, g)^{\alpha_1 z} \tag{7}$$

- *sa* recovers the item key as $ik = C_1 / e(g, g)^{\alpha_1 z}$, and then obtains E_w by using AES decryption.

Afterwards, *sa* conducts detailed analysis on E_w, and sends an analysis report or some newly recovered evidence to *pi* and receives a signature.

5.7. Court trial and subsequences

5.7.1. Court trial

During a court trial, the prosecution lawyer *pl* issues a charge and the defense lawyer *dl* plead for the accused. Then both send corresponding data to the judge *jg* and upload a hash value for their data to the blockchain. Jurors send their vote to the blockchain network via the slicing approach. The idea of the slicing approach is that jurors slice their votes into three subvotes and send an encrypted subvote to each blockchain miner. The blockchain miners collect all subvotes, multiply them to get an aggregate ciphertext, and decrypt the partial result to obtain a partial vote result. Then the sum of three partial vote results is the final vote result.

Before the judge announces a trial result, each juror jr_i ($1 \le i \le 12$) submits a vote as follows:

- Generate an original vote $\mathcal{V}_i \in \{0, 1\}$, split it into three slices $\mathcal{V}_{i1}, \mathcal{V}_{i2}, \mathcal{V}_{i3}$ satisfying $\mathcal{V}_i^1 + \mathcal{V}_i^2 + \mathcal{V}_i^3 = \mathcal{V}_i$.
- Encrypt $\mathcal{V}_{i1}$, $\mathcal{V}_{i2}$ and $\mathcal{V}_{i3}$ using ElGamal encryption [35] with pk_{pd}, pk_{co}, and pk_{pr} respectively. Obtain three encrypted votes $\mathcal{V}'_{i1}$, $\mathcal{V}'_{i2}$, and $\mathcal{V}'_{i3}$. In this way, the juror has protected the vote privacy.
- Generate an anonymous credential on the three encrypted votes using anonymous authentication as the witness did and cast it to the blockchain via a transaction.

Upon receiving all votes from the twelve jurors via the blockchain, *pd* aggregates its encrypted votes by computing $\Pi_{i=1}^{12} \mathcal{V}'_{i1}$ and decrypts the aggregate ciphertexts to obtain a partial vote result V_{pd}. Next, *pd* broadcasts V_{pd} with a signature to the blockchain network. *co* and *pr* perform the same for the aggregate vote results $\Pi_{i=1}^{12} \mathcal{V}'_{i2}$ and $\Pi_{i=1}^{12} \mathcal{V}'_{i3}$, and two partial vote results V_{co} and V_{pr}. The final vote result is $V_{final} = V_{pd} + V_{co} + V_{pr}$.

The judge first verifies V_{final} by checking the $\{\mathcal{V}'_{i1}, \mathcal{V}'_{i2}, \mathcal{V}'_{i3}\}$ with $\Pi_{i=1}^{12} \mathcal{V}'_{i1}$, $\Pi_{i=1}^{12} \mathcal{V}'_{i2}$, and $\Pi_{i=1}^{12} \mathcal{V}'_{i3}$. By doing so, the vote result is verifiable. If V_{final} is valid, *jg* generates a trial result *Res* based on V_{final}, preserves court data *CD* in the court, computes a hash value of $Res \parallel CD$, generates a signature, and sends a transaction including *Res*, hash value, and signature to the blockchain. The general process of the first voting method is shown in Fig. 3.

5.7.2. Other transactions

The prison detains prisoners and keeps prisoners' records, e.g., imprisoned, transferred, and released. The hospital treats wounded entities and uploads treatment histories, including active treatment, hospitalized, and healed. The repair factory repairs partially damaged vehicles/devices and uploads repair records, e.g., beyond repair, in repair, and fixed. The insurance company compensates entities according to the judgment and company policies, and uploads insurance records, including beneficiary and amount of money. All of them upload a hash value of their data to the blockchain network.

The police department, court, and prison co-maintain the *CBC* by running the PoA consensus mechanism and packs a new block containing a block header, all the uploaded hash values, and corresponding signatures during each time slot.

5.7.3. Entity tracking

While the blockchain can provide the integrity and auditability of evidence and court data, some malicious entities can launch data injection, data falsification attacks, and unauthorized access attacks. The identity of such entities must be traced in time. For an anonymous entity which submitted a signature before, the *TA* looks up all items $(i, \tau_i, \eta_i, \hat{\tau}_i)$ in its database and checks whether $e(\sigma_2, g) \cdot e(\sigma_1, \hat{X})^{-1} = e(\sigma_1, \tau)$ until one match is found. Then the *TA* outputs (i, τ_i, η_i) along with a proof of knowledge of a valid $\hat{\tau}_i$.

5.7.4. Attribute revocation

When an entity is removed from a case or degraded in the system, some of his attributes should be removed. Suppose a police investigator *pi* is leaving a case and his attribute x should be revoked so that he cannot access related evidence stored in the police department *pd*. We use non-revoked entities to denote entities who still possess x.

The *TA* selects a random number $v'_x \in \mathbb{Z}_p$ ($v'_x \neq v_x$) as a new attribute version key, computes an update key $uk_x = (uk_x^1 = v'_x / v_x, uk_x^2 = (v_x - v'_x)/(v_x \alpha_3))$, sends uk_x to all the non-revoked entities and *pd* via secure channels, renews the public attribute key of x as $pak'_x = (pak_x^1 = \mathcal{H}(x)^{v'_x}, pak_x^2 = \mathcal{H}(x)^{v'_x \alpha_3})$, and broadcasts a message including pak'_x that the public attribute key of the revoked attribute x is renewed. Next, each non-revoked entity sends $L = g^o$ and S_x to the *TA* which computes a new S'_x as $S'_x = (S_x / L^{\alpha_2^2})^{uk_x^1} \cdot L^{\alpha_2^2} = g^{o \alpha_2^2} \cdot \mathcal{H}(x)^{v'_x o \alpha_2}$ and returns it to the non-revoked entity. The entity's user secret key is updated as $usk' = (S, L, S'_x, \forall x \in A \backslash \{x\} : S_x)$. Finally, *pd* renews the ciphertext associated with x to obtain a new C_{pd}^x.

6. Security and privacy analysis

6.1. Authentication

Definition 1 (*Lysyanskaya–Rivest–Sahai–Wolf (LRSW) Assumption 1*). Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ be a bilinear group setting of type 3. g and $\hat{g}$ are two generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. For $x, y \in \mathbb{Z}_p, X = g^x, Y = g^y, \hat{X} = \hat{g}^x, \hat{Y} = \hat{g}^y$, we define an oracle $\mathcal{O}(i)$ on input $i \in \mathbb{Z}_p$ that randomly chooses a number $r \in \mathbb{G}_1$ and outputs a signature $\sigma = (r, r^{x+my})$. Given $(g, Y, \hat{g}, \hat{X}, \hat{Y})$ and access to $\mathcal{O}$, no adversary is able to efficiently produce a new $(r, r^{x+i'y})$ with $(r \neq 1_{\mathbb{G}_1})$ for a new i' which has not been queried to $\mathcal{O}$ [29].

Theorem 1. *The existential unforgeability under chosen message attacks of an entity's signature holds under the LRSW assumption 1.*

Proof. Let Π' be the signature scheme in LRSW assumption 1, and Π be our proposed LEChain. Let $\mathcal{A}$ be a probabilistic polynomial-time (PPT) adversary breaking Π with $q = q(k)$ an upper bound on the number of queries that A makes to the $\mathcal{O}$. $\mathcal{A}$ makes any

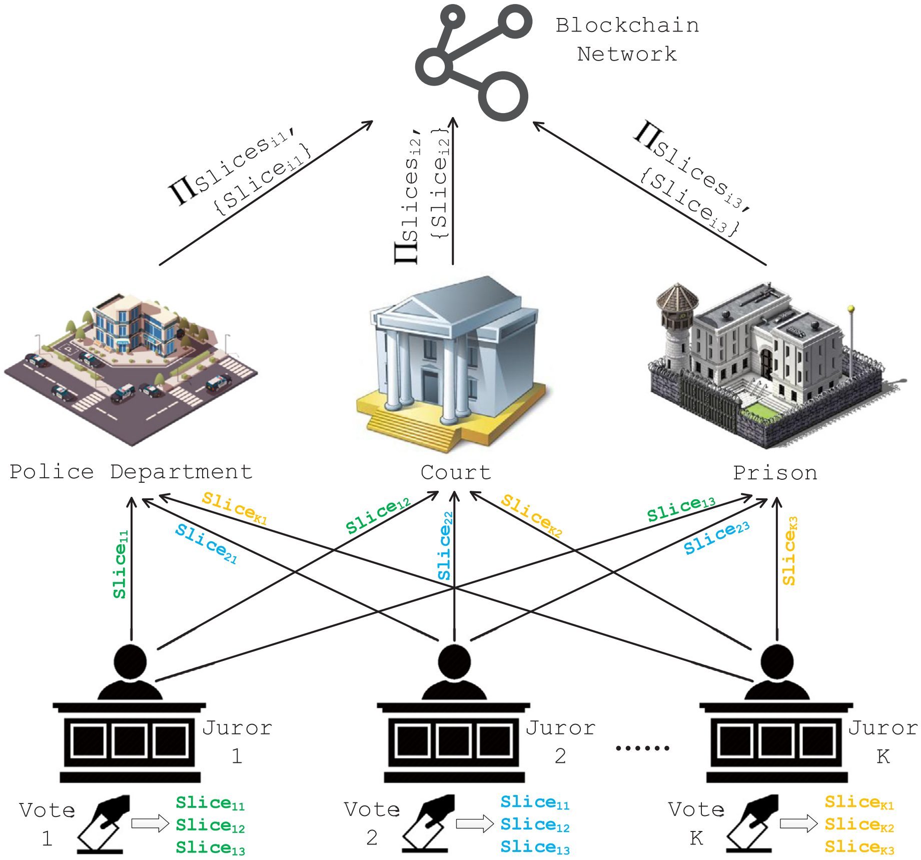


Fig. 3. The illustration of jurors cast votes via slicing.

query to $\mathcal{O}$ only once. $\mathcal{A}$ never queries m after being given a valid signature $\sigma = (\sigma_1, \sigma_2)$ on m where $e(\sigma_1, \hat{X} \cdot \hat{Y}) = e(\sigma_2, \hat{g})$. $\mathcal{A}$ outputs a forged signature σ' on a new message m' if Verify$(pk, m', \sigma') = 1$, i.e., $e(\sigma_1', \hat{X} \cdot \hat{Y}^{m'}) = e(\sigma_2', \hat{g})$. Now we construct an efficient adversary $\mathcal{A}'$ that calls $\mathcal{A}$ and tries to break Π'.

Algorithm $\mathcal{A}'$**:** $\mathcal{A}'$ is provided with pk in Π including public parameters $pp_1 \leftarrow (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ and public key $pk = (\hat{g}, \hat{X}, \hat{Y})$ and access to $\mathcal{O}$.

1. Choose uniform $i \in \{1, 2, \ldots, q\}$.
2. Call $\mathcal{A}(pp_1, pk)$. If $\mathcal{A}$ makes the jth query m_j, responds as follows: if $j = i$, query $\mathcal{O}$ to obtain a transcript (m, σ_1, σ_2) of a normal execution of LEChain and return (sig_1, sig_2); if $j \neq i$, choose uniform $\sigma_1, \sigma_2 \in \mathcal{G}_1$ and return (σ_1, σ_2).
3. If $\mathcal{A}$ outputs a forged signature (σ_1', σ_2') on a new message m', check $e(\sigma_1', \hat{X} \cdot \hat{Y}^{m'}) \overset{?}{=} e(\sigma_2', \hat{g})$ and $m' \neq m_j$. If yes, output $(m', \sigma_1', \sigma_2')$. If no, abort.

The view of $\mathcal{A}$ when called as a subroutine by $\mathcal{A}'$ in the experiment Sig-forge$_{\mathcal{A}',\Pi'}$ is the same as the view of $\mathcal{A}$ in the experiment Sig-forge$_{\mathcal{A},\Pi}$. If $e(\sigma_1', \hat{X} \cdot \hat{Y}^{m'}) \overset{?}{=} e(\sigma_2', \hat{g})$ and $\mathcal{A}'$ never queries m' to $\mathcal{O}$, $(m', \sigma_1', \sigma_2')$ is a valid signature forgery.

Hence, we now conclude that Pr[Sig-forge$_{\mathcal{A}',\Pi'}$] $\geq \frac{1}{q(k)} \cdot$Pr[Sig-forge$_{\mathcal{A},\Pi}$]. If Π' is secure, then Pr[Sig-forge$_{\mathcal{A}',\Pi'}$] is negligible. Since $q(k)$ is polynomial, it indicates that Pr[Sig-forge$_{\mathcal{A},\Pi}$] is negligible as well. The proof is complete. □

6.2. Privacy

First, the witness w utilizes an anonymous credential [29] to prove her/his qualification of uploading evidence in the system to prevent the police investigator pi and any other entities from obtaining the real identity of the witness. If a witness uploads some evidence twice, she/he will randomize the credential in each evidence upload such that any two of them cannot be linked. Hence, witness privacy is guaranteed due to the anonymity and unlinkability above. Second, before a juror casts a vote to the blockchain, she/he will use a similar anonymous credential to protect her/his real identity. Each original slice is split into three slices (please refer to the slicing mechanism detailed in Section 5.7) which are sent to the police department, court, and prison using ElGamal encryption, separately. Since the ElGamal encryption is semantically secure against chosen plaintext attacks [35] based on the Hash–Diffie–Hellman problem, only the three authorities can decrypt the received slices. Even so, the three authorities can only decrypt one slice of each juror and do not know the other slices of each juror. Given that the three authorities do not collude with each other, each juror's original vote remains private. Therefore, juror privacy is guaranteed. Third, data privacy is protected because we used a fine-grained access control mechanism to prevent unauthorized access against malicious police investigators. Please refer to the next subsection for a detailed proof.

6.3. Access control

If an entity which does not possess enough attributes satisfying the access structure $\mathcal{T}$ of evidence E tries to access E, she/he cannot reconstruct the encryption exponent z, let alone decrypt the ciphertext of E with her/his user secret key. If one or some attributes of an entity are revoked, and the entity tries to access the data with the old user secret key, the access will be unsuccessful. For example, let us assume that an attribute x is revoked from an entity, the TA selects a new attribute version key to produce a new update key and sends it to the data preservation authority to update all the ciphertexts related to x. Due to the different values of the attribute version key in the ciphertext, the revoked entity cannot decrypt the ciphertext with the old user secret key. Therefore, LEChain defends the unauthorized access attack and provides access control.

6.4. Integrity & auditability

Before the hash value of data is uploaded to the blockchain, it has to be signed by a data uploader, e.g., police investigator and judge. After the hash value is stored on the blockchain, the unforgeability of the blockchain has guaranteed the data integrity. If some evidence is handled improperly, such as a malicious police investigator has falsified the evidence, this behavior could be detected due to two reasons. First the data is signed by a data uploader and the signature is held by the data provider who can verify data integrity. Second, the hash value of the encrypted data is stored on the blockchain which is publicly verifiable. After the data is falsified, the data provider points out this inconformity by showing the data, encrypted data, and the signature. Even if a malicious entity launches data falsification attacks, the data providers, e.g., victims and witness, will present the original data to invalidate the falsified data. Meanwhile, if a malicious entity injects some forged data, there will be no real data providers to validate this source.

Each entity which has uploaded some evidence or court data is able to audit the blockchain data via computing a hash value of their own data and comparing the hash value with the data on the blockchain. Therefore, the integrity and auditability are guaranteed.

In practice under some circumstances, the judge deviates from the result of the jurors. This misbehavior could be detected by comparing the consistence between the trial result *Res* and the V_{final}. This is possible because (1) *Res* and V_{final} are recorded on the blockchain, (2) V_{final} is verifiable since all the votes from the twelve jurors and the partial vote results from the blockchain nodes are recorded on the blockchain. Any entity can fetch them from the blockchain and reconstruct V_{final}.

6.5. Traceability

In evidence upload, there are two sources for the evidence E which are evidence provider (e.g., victim vc and witness w) and police investigator *pi*. The source of E can be traced because (1) E provided by vc includes the identity of vic and it is signed by *pi*. The owner of the signature can be recovered by the *TA*. (2) E provided by w is tied to a group signature $\mu_w = (\hat{\sigma}'_{w1}, \hat{\sigma}'_{w2}, ch_w, ss)$ and signed by *pi*, and the *TA* can open μ_w and trace the identity of w by testing $e(\hat{\sigma}'_{w1}, \hat{g}) \cdot e(\hat{\sigma}'_{w1}, \hat{X}) \overset{?}{=} e(\hat{\sigma}'_{w2}, \hat{\tau})$ through all database items $(i, \tau_i, \eta_i, \hat{\tau}_i)$.

In data access, the sources of the evidence are crime scene analyst *sa* and police investigator *pi*, they can be traced by *sa*'s group signature and *pi*'s signature as well.

We present Table 2 to compare LEChain with existing works regarding security and properties. Witness privacy is protected for most schemes because of BBS signatures in [7], k-anonymity techniques in [8], pseudonym certificates from IEEE 1609.2 standard in [2], and Merkle signatures in [6]. "n/a" in the Table means that this item is not applicable to the scheme because it does not include corresponding role in its system model. Other than LEChain, only [4] provides access control by using access control rules in Hyperledger composer. Traceability is provided since the group manager can recover the real identity of group members in [7], pseudonym certificates are recorded in a certificate authority in [2], and requiring the file name include the provider's name, i.e, IP address in [9].

Table 2
Comparison of security and privacy properties.

Property	[7]	[8]	[2]	[6]	[4]	[9]	[23]	LEChain
Witness privacy	✓	✓	✓	✓	n/a	n/a	×	✓
Juror privacy	n/a	n/a	n/a	n/a	n/a	n/a	n/a	✓
Authentication	✓	✓	✓	×	✓	✓	✓	✓
Access control	×	×	×	×	✓	×	×	✓
Integrity	✓	✓	✓	✓	✓	✓	✓	✓
Auditability	✓	✓	✓	✓	✓	✓	✓	✓
blueTraceabiity	✓	×	✓	×	×	✓	×	✓

7. Performance analysis

7.1. Experiment settings

We instantiated 100 victims, 100 witnesses, 100 crime scene analysts, ten police investigators, one police department, one court, one prison, 12 jurors, and one judge on a laptop with 8.00 GB of RAM, an Intel Core i7-7500 CPU @2.70 GHz, running Windows 10 Home and Visual Studio 2010. The victims and witnesses provided evidence more than once, and the crime scene analysts applied for evidence access repeatedly. Uploading and accessing transactions are randomly generated. The used cryptographic toolset was Miracl.[4] The elliptic curve was defined as $y^2 = x^3 + 1$ over $\mathbb{F}_q$. We chose Ethereum as the blockchain platform and the Clique, a kind of PoA, as the consensus mechanism. In Clique, there are two kinds of nodes: authenticated node and unauthenticated node. The former one has the right to sign while the latter one does not. The two kinds of nodes can switch to each other. Clique has achieved a voting mechanism to bring in or exclude signer. Meanwhile, the authenticated nodes use a fair node creation approach to control the creation time of a new node. The new block creation time was set to 10 s. We note that we use a consortium blockchain to record all the evidence related transactions. The TA, the police department, the court, and the prison are the four blockchain miners. We recommend at least four operators are needed at a minimum to avoid security risks. Specifically, we first installed geth (mini server) and MetaMask (communicated with Geth via Http), and created four virtual machines (three Ubuntu 18.04 and one Windows 7) on vmware. Then we created a file folder "node" and signature account. Next, we used puppeth to generate the genesis block. After peer authorization and configuration, we used MetaMask on the browser to import a key file opened the virtual machines to initiate geth. The smart contract is written on Remix. Specifically, a smart contract is a program or scripts stored on the blockchain and it has a unique address. The contract is triggered by receiving transaction(s) and executes automatically on every node in the network according to the triggering transaction. A finite state machine can describe the smart contract and each state denotes a global state of the investigation. There are seven states: collection, upload, upload rejected, store, access, vote, and completed. Different actions trigger the state transition. Each state transition is attached with a digital signature in a transaction and verified by multiple blockchain nodes. We present the LEChain smart contract in Table 3. The key experimental parameters are listed in Table 4. The source codes are uploaded to GitHub.[5]

[4] http://www.certivox.com/miracl

[5] https://github.com/SopmmmodII/LEChain

Table 3
LEChain smart contract.

On receiving (“**Request**”, pi, ts) from pi
Create CurrentState = Collection;
Insert collection request pool $CRP[\mathcal{H}(pi, ts)] = (pi, ts)$;
Broadcast (“A collection request has been created.”);
On receiving (“**Upload**”, pi, R_w, $\mathcal{H}(c_w)$, ts, ρ^{w}_{pi})) from pi
Set CurrentState = Upload;
Put $CRP[\mathcal{H}(pi, ts)].dataHash = \mathcal{H}(c_w)$;
Broadcast (“New evidence have been collected by a police investigator.”);
On receiving (“**Reject_Upload**”, Tx_{up}, sig) from a block node
Delete $CRP[\mathcal{H}(pi, ts)]$;
Set CurrentState = Upload Rejected;
Broadcast (“The upload has been rejected by a blockchain node.”);
On receiving (“**Store**”, $\mathcal{H}(c_w)$, ts_{pd}, sig) from pd
Set CurrentState = Store;
Put $CRP[\mathcal{H}(pi, ts)].downloadLink = \mathcal{H}(c_w \vert\vert ts_{pd})$;
Broadcast (“New evidence have been stored.”);
On receiving (“**Access**”, sa, $\{c^{i}_{sa}\}$, ts_{sa}, ρ_{sa}) from sa
Set CurrentState = Access;
Broadcast (“An access has been granted.”);
On receiving (“**Vote**”, $\mathcal{V}'_1$, $\mathcal{V}'_2$, and $\mathcal{V}'_3$) from jr
Set CurrentState = Vote;
Broadcast (“A vote has been submitted.”);
On receiving (“**Complete**”, V_{final}, sig) from jg
Set CurrentState = Completed;
Broadcast (“A trial result has been achieved.”);

Table 4
Key experimental parameters.

Parameters	Value
CPU	Intel Core i7-7500, 2.70 GHz
RAM	8.00 GB
Operating system	Windows 10 Home
$\vert p\vert$, $\vert q\vert$	160,512
$\mathcal{H}$	$SHA256$
$\mathcal{V}$	{0, 1}
l, n, $\vert ik\vert$	10, 5, 256
$\vert\{vt\}\vert$, $\vert\{w\}\vert$, $\vert\{sa\}\vert$, $\vert\{pi\}\vert$	100, 100, 100, 10
$\vert\{pd\}\vert$, $\vert\{co\}\vert$, $\vert\{pr\}\vert$	1, 1, 1

7.2. Computational costs

We first analyzed the computational costs for the main entities: witness w, crime scene analyst sa, police investigator pi, police department pd, juror ju, and judge jg through counting the total number of cryptographic operations.

We use the following terms $E_1, E_2, E_T, M_1, M_T, D_1, D_T, BP$, and $\tilde{A}, \tilde{S}, \tilde{M}, \tilde{D}, \tilde{E}$ as exponentiation in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$, multiplication in $\mathbb{G}_1, \mathbb{G}_T$, division in $\mathbb{G}_1, \mathbb{G}_T$, bilinear pairing, addition, subtraction, multiplication, division, exponentiation in Z_p. We give the measured results for the implemented running time of each entity listed in Table 5.

In system initialization, the TA computes a group public key for anonymous authentication ($2E_2$), four public parameters for access control ($3E_1 + \hat{D} + BP$), and a set of public attribute keys ($l * (2\mathcal{H} + \hat{M} + 2\hat{E})$), which takes approximately 21.1 ms. In entity registration, both w and $juror$ obtains a group secret key ($3E_1 + E_2 + \hat{M} + \hat{A}$) which takes approximately 10.8 ms, the TA assists them in generating a group secret key ($2BP + 4E_1 + M_1$). TA computes a user secret key for sa ($\hat{A} + 4\hat{M} + 2\hat{E} + \hat{D} + 7E_1$). sa has 5 attributes.

In evidence collection, pi generates a signature and a hash value for vt ($E_1 + \mathcal{H}$) and w prepares a witness report ($4E_1 + M_1 + \hat{E} + E_T + \mathcal{H} + \hat{M} + \hat{A}$) where $e(\sigma_{w1}, \hat{Y})$ can be pre-computed. In evidence upload, pi verifies a witness report ($2E_1 + 3BP + \hat{S} + E_T + 2M_T$), generates a signature (E_1), decrypts the evidence ($E_1 + D_1$), and prepares an upload transaction ($\mathcal{H} + E_1$). The time cost for Verify here is the verification performed by a blockchain node towards an upload transaction, which mainly include anonymous signature verification. In evidence access, pd encrypts evidence ($n * AESEnc + BP + E_T + (n^2 + 6)\hat{M} + 5E_1 + 2\hat{S} + 2\mathcal{H} + \hat{E} + M_1 + \hat{D}$). sa forms an access transaction ($1\hat{M} + 4E_1$) and decrypts the evidence ($BP + l * (2BP + M_T + E_T) + D_T + AESDec$).

In court trial, jr encrypts a vote and an anonymous credential ($4E_1 + M_1 + \hat{M} + E_T + \mathcal{H} + \hat{M} + \hat{A}$), the three authorities aggregate slices, decrypt the aggregate result and generate a signature ($11M_1 + 2E_1 + D_1 + 2\hat{A}$). jd verifies the final vote result ($3 * (11E_1 + E_1 + D_1) + 2\hat{A}$). In identity tracking, the TA computes $3BP + E_T$ for one database item. In attribute revoking, the TA computes an update key ($2\hat{D} + \hat{S} + \hat{M}$) and renews the public attribute key ($2\mathcal{H} + \hat{M} + \hat{E}$). TA assists sa in computing a new S'_x ($2\hat{E} + 3E_1 + D_1 + M_1$). pd renews the ciphertext ($M_1 + 2E_1$).

We present the time cost in evidence collection, evidence access, and attribute revoking in Figs. 4(a), 4(b), and 4(c). The experimental results show that the time cost of entities grows linearly with the number of evidence and the number of revoked attributes. It takes approximately less than 1 s to process all the data. The police department spends more time in evidence access because we use fine-grained access control, and the police department has to encrypt all the evidence.

7.3. Communication overhead

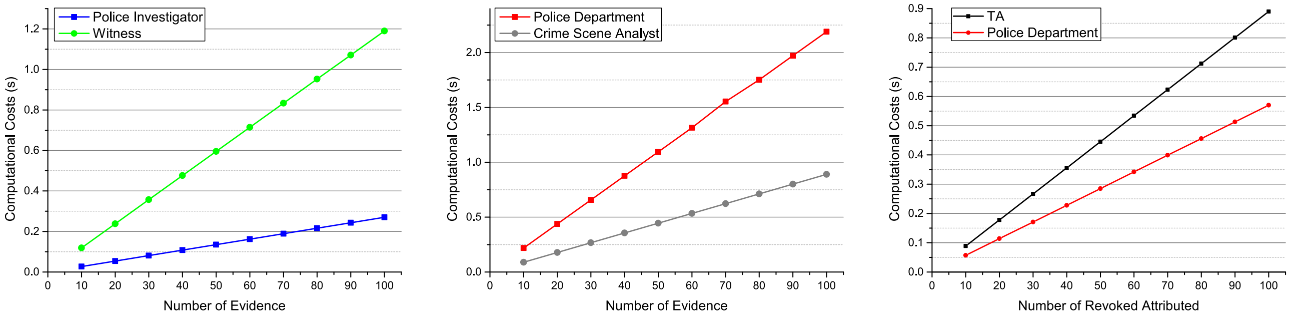
Then we analyzed the communication overhead for the main entities. The witness w sends a witness report R_w which has a bit length of $|c_w| + |\hat{\sigma}'_{w1}| + |\hat{\sigma}'_{w2}| + |ch_w| + |ss| = 160 * 2 + 160 + 160 + 256 + 160 = 0.129$ KBytes. For one piece of evidence, the police investigator pi returns a signature to the evidence source and submits an upload transaction Tx^{up}_{pi}. The total size is $|$“Upload"$| + |pi| + |R_w| + \mathcal{H}(c_w)| + |ts| + |\rho^{w}_{pi}| = 0.200$ KBytes. The crime scene analyst sa submits an access transaction Tx^{ac}_{sa} and the communication overhead is $|$“Access"$| + |sa| + |\{c^{i}_{sa}\}| + |ts_{sa}| + |\rho_{sa}| = 5 + 7 + 3 * 2 * 160 + 10 + 160 = 0.139$ KBytes. The police department pd sends the encrypted evidence to a querying entity and broadcasts a partial vote result with a signature in a court trial. The total communication overhead is 0.129 KBytes. Similarly, the juror ju and the judge jg need to send 0.051 KBytes and 0.387 KBytes of data, respectively.

We deployed four miners in the blockchain network and allowed a user (police investigator) to send an upload transaction 40 times. During this process, we tested the transaction confirmation time by using waitForTransactionReceipt() and recorded elapsed time by using time.time(). The programming language is python3.6 with the library web3py. The results are illustrated in Fig. 5(a). The average confirmation time for an upload transaction is 10 s, i.e, each upload transaction needs 1 block time to be finally confirmed in the blockchain.

We also tested the network latency in accessing evidence, i.e., the time elapsed from a crime scene analyst' submitting an access transaction to the crime scene analyst' receiving a download link from the blockchain. The results are illustrated in Fig. 5(b). The average network latency is approximately 2.5 ms.

Table 5
Implemented running time (Unit: Millisecond).

Phase	Initial.	Registration			Collect		Upload	Verify	Access	
Entity	*TA*	*w*	*jr*	*TA*	*pi*	*w*	*pi*	node	*sa*	*pd*
Time	21.1	10.8	10.8	48.7	2.7	11.9	26.4	24.3	8.9	21.9
Phase	Court Trial					Tracking			Revoking	
Entity	*jr*	*pd*	*co*	*pr*	*jd*	*TA*			*TA*	*pd*
Time	11.9	8.9	8.9	8.9	26.7	20.9			8.9	5.7

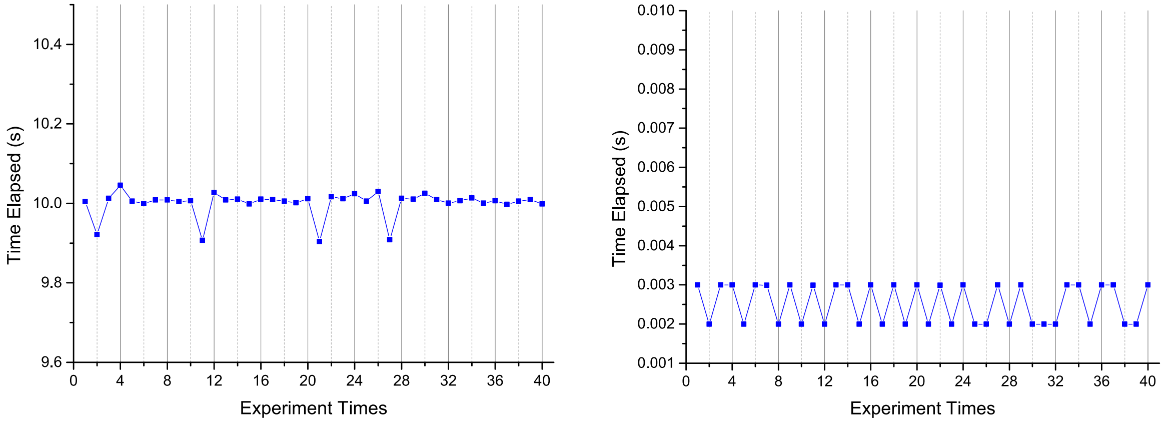


(a) Time Cost in Evidence Collection

(b) Time Cost in Evidence Access

(c) Time Cost in Attribute Revoking

Fig. 4. Computational costs.



(a) Upload Transaction Confirmation Time

(b) Network Latency in Accessing Evidence

Fig. 5. Communication overhead.

7.4. Comparison to existing work

Since the system model varies with different systems, we first focus on comparing with existing work regarding computational costs in main operations or phases: upload, verification, and access. The results are recorded in Table 6. The computational cost is high in the work of Zhang et al. [7] for their adoption of BBS signatures in uploading and verifying evidence which incurs time consuming operations. Nieto et al. [8] only use k-anonymity to protect the anonymity of evidence uploader which does not involve cryptographic primitives. It takes less than 1 ms for [2] to upload and verify evidence because it only leverages pseudonym certificates to guarantee privacy. The time cost of [6] is similar to [2] for using Merkle signatures which are similar to pseudonym certificates. Lone et al. [4] utilize Ethereum to enable evidence management which uses ECDSA in evidence upload and verification. Although Lone et al. [4] provide access control in their system, they do not mention which specific mechanism or method is used. Tian et al. [9] have facilitated 2-of-2 multi-signatures (from Ohta and Okamoto multisignature) in uploading and verifying evidence. In the work of Frankle et al. [23], both an evidence uploader and an evidence requester have to compute a commitment and a zero-knowledge proof which results in a relatively high cost.

Next, we focus on comparing with existing work regarding communication overhead in main operations or phases: upload and access. The results are recorded in Table 7. Although LEChain does not perform the best among all the schemes, it achieves the adequate security and privacy protection. Briefly, the LEChain provides authentication and traceability by using group signatures. It protects privacy via leveraging anonymous authentication. It also achieves fine-grained access control based on ciphertext-policy attribute-based encryption. The integrity and auditability are guaranteed for using the blockchain technology.

Table 6
Comparison of computational costs (Unit: Millisecond).

Operation	[7]	[8]	[2]	[6]	[4]	[9]	[23]	LEChain
Upload	45.8	0.8 ($k = 10$)	0.2	0.2	3.0	0.2	1.2	26.4
Verification	55.2	n/a	0.2	0.2	5.4	0.2	60	24.3
Access	n/a	n/a	n/a	n/a	n/a*	0.2	61.2	30.8

Table 7
Comparison of communication overhead (Unit: KBytes).

Operation	[7]	[8]	[2]	[6]	[4]	[9]	[23]	LEChain
Upload	0.624	2.19	0.219	0.219	0.16	0.25	0.5	0.329
Access	n/a	n/a	n/a	n/a	n/a	0.25	0.5	0.268

8. Discussion

8.1. Evidence analysis

For evidence analysis within the police department, *sa* conducts detailed analysis of some evidence after obtaining it from the police department. When the analysis is complete, *sa* will send a report or newly discovered evidence to a police investigator and receive a signature. For other types of analysis, such as analysis on contaminated water relying on some professional equipment, police investigators resort to specific institutions or experts. Such institutions participate in the blockchain network by registering to the *TA* and obtaining corresponding keys.

8.2. Evidence transfer

Some different police bureaus affiliating to the police department do need to transfer evidence among themselves. They can achieve this with the help of our consortium blockchain. For instance, police bureau A transfers one photo P of a suspect to police bureau B by sending a transferring transaction Tx^{tr}_{pbA} = ("Transfer", pb_A, pb_B, $\mathcal{H}(P)$, ts_{pb_A}, ρ_{pb_A}) and transports the original photo to B with armed police escort.

8.3. Evidence disposition

If some evidence E is no longer valid, the police department can broadcast a disposition transaction ("Disposition", $\mathcal{H}(E)$, pd, Tx^{up}_{pi}, ts_{pd}, ρ_{pd}) indicating that this evidence is not relevant to the police investigation. Here, evidence disposition does not mean deleting actual evidence, but removing corresponding information from the blockchain via nullifying a previous upload transaction Tx^{up}_{pi}. Since the "Transfer" function and "Disposition" function are not the main part of LEChain, we consider them in the future work.

8.4. The contradiction between witness privacy and traceability

We protect witness privacy in the evidence collection phase. But it is a reality that a judge or jurors need to meet the corresponding witness on the court and further verify the evidence. Even so, the identity of the witness must be protected by the witness protection programmes [36–38]. In Europe, such programmes include video conferencing with voice and face distortion are used to hide a witness' identity. Some threatened witnesses might also be protected by police accompanying officers and patrolling police vehicles [36]. In Canada, the Royal Canadian Mounted Police have provided both short-term measures (e.g., immediate responses to potential threats) and long-term measures (e.g., relocation, change of identity, counseling, and financial support) [37]. Similarly, the U.S. Marshals have protected, relocated, and issued new identities to over 8,600 witnesses, which have been widely considered as exerting a positive effect on the battle against criminal activities [38].

8.5. Data privacy in long term and post-quantum crypto

Long-term data activities gradually increase identifiability and amplify the harms to which users are exposed, and the key components are the amount of time after collection, the length of the time period within which data are repeatedly measured, and the time period between reiterant measures on the piece of data [39]. To solve this problem, we could adopt the update functionality in [40] which is a protocol for inserting or removing an entry from the database. During the updates, the requests that the storage server receives are indistinguishable from random because of the pseudorandomness of pseudorandom function. In this way, the server does not know the type of operation, and the update leakage is none. Other solutions include enforcing multiple layers of data protection, such as explicit consent, statistical disclosure control, and data use agreements [40].

Based on quantum physics, quantum computing uses quantum bits to break through the limitation on the speed of today's computers. The modern public key cryptography is based on hard mathematical problems, such as factorization for RSA algorithms, discrete logarithm problems, and Diffie–Hellman problems. But an adversary will weaken them by using quantum algorithms if he has a quantum computer. Currently no quantum computer is able to run quantum algorithms, but there are indicators that it will happen after 2023 [41]. If this happens, a post-quantum data risk assessment must be undertaken in advance to consider updating crypto policies, classifying data flows, and creating a timeline for quantum safe crypto [42]. Afterwards, corresponding post-quantum implementation strategies and schemes must be developed in order to protect quantum-safe security, which has three pillars: quantum random number generators, quantum key distribution, and a set of quantum resistant algorithms. Conclusively, post-quantum protection mechanisms will be designed for data privacy in post-quantum crypto era.

8.6. Consensus mechanism

We chose Ethereum as the blockchain platform, and the Clique, a kind of PoA, as the consensus mechanism. In Clique, there are two kinds of nodes: authenticates node and unauthenticated node. The former one has the right to sign while the latter one does not. The two kinds of nodes can switch to each other. Clique has achieved a voting mechanism to bring in or exclude signer. Meanwhile, the authenticated nodes use a fair node creation approach to control a new node's creation time. However, the blockchain is not "secure" since the PoA is vulnerable to the "Cloning Attack" [43] where a sealer adversary cloning a private key to persuade half of the honest sealers that a transaction is properly committed before erasing this transaction to double spend its coins. But countermeasures can be implemented [43].

8.7. Distributed storage systems

In IPFS, the data files are stored and replicated on many computers, and it is a good option while using off-chain data with one way linking to data through blockchain. However, it is not relevant to the criminal investigation issues that we address in this work. Therefore, we only use IPFS [44,45] in our experimental analysis. While in real world implementation, the choice of distributed storage systems is extremely crucial. Commercial distributed storage is not recommended for digital forensics, especially when it involves police investigation. Hence, the police department, court, and other lawful sections should be collaborating to establish their own distributed storage systems for storing evidence.

8.8. Scalability

We acknowledge that scalability is an important aspect for a system running in real world. However, in this work, we mainly focus on modeling a complete chain of evidence for digital forensics as well as its security and privacy protection. In the experiments, we realized a prototype of LEChain and show its feasibility. The given experimental results do not undermine the scalability. The determinant factors of scalability lie in the capability of the blockchain nodes and distributed storage system. If they can provide computation and communication capabilities strong enough to perform the message broadcasting, data verification, and data retrieval, the scalability will speak for itself. Some blockchain applications have already shown their scalability in real-world implementation, such as Bitcoin, Ethereum, and Hyperledger Fabric [12,46]. But still, it remains a future work for us to enhance LEChain's practicality by integrating it with real-world investigation. We have leveraged Ethereum and PoA to build the prototype and test its performance. But still, we are planning to realize real world deployment in future work.

8.9. Number of jurors

We use twelve as the number of jurors in the LEChain to as an example to be applied in the US justice system. However, this number is a generic value so the LEChain can be applicable to more justice systems. For example, it is 8 at the start of a trial in England and Wales; it is 8 at regional courts in Russia.

9. Conclusion

In this paper, we have analyzed the potential security and privacy threats in lawful evidence management for digital forensics. We have proposed a lawful evidence management scheme named LEChain on top of blockchain and smart contracts. In LEChain, the data first involves the entire evidence flow from the collection, to examination, to analysis and reporting. Then court data are also included. LEChain achieves efficient management, i.e., unforgeability and auditability, of evidence and court data with the assistance of a consortium blockchain. LEChain guarantees fine-grained access control of evidence and court data based on ciphertext-policy attribute-based encryption. It protects witness privacy during police investigation and protects juror privacy during a court trial. With LEChain, lawful entities can better upload and access corresponding data via the blockchain which prevents malicious police investigators from falsifying the evidence. Witnesses and jurors can feel free and secure to participate in the system. Finally, we have implemented the proposed scheme on the Ethereum public test network to test its feasibility and efficiency.

In future work, we will enhance the security protection on lawful evidence by considering data leakage attacks and designing detection mechanisms [47]. The data leakage attack refers to a police investigator intentionally leaks her/his gathered evidence to illegal entities or sells the evidence to black market. Defending against this attack will protect the evidence from being leaked on social media [48] and further protect the evidence chain. In the experiment, we realized a prototype of LEChain and showed its feasibility. But still, it remains a future work to enhance its practicality and scalability by integrating it with real-world investigations. Also, we put the functions "Transfer" and "Deposition" in future work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by Anhui Provincial Natural Science Foundation under the grant No. 2008085MF196, National Natural Science Foundation of China (NSFC) under the grant No. 62002094, Anhui Science and Technology Key Special Program under the grant No. 201903a05020016, and National Natural Science Foundation of China (NSFC) under the grant No. U1836102. It is partially supported by EU LOCARD Project under Grant H2020-SU-SEC-2018-832735.

References

[1] A. Nieto, R. Rios, J. Lopez, Digital witness: Safeguarding digital evidence by using secure architectures in personal devices, IEEE Netw. 30 (6) (2016) 34–41.

[2] M. Cebe, E. Erdin, K. Akkaya, H. Aksu, S. Uluagac, Block4forensic: An integrated lightweight blockchain framework for forensics applications of connected vehicles, IEEE Commun. Mag. 56 (10) (2018) 50–57.

[3] X. Lin, T. Chen, T. Zhu, K. Yang, F. Wei, Automated forensic analysis of mobile applications on Android devices, in: Proc. 18th Annual DFRWS USA, Rhode Island, USA, 2018, pp. 59–66.

[4] A.H. Lone, R.N. Mir, Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer, Digit. Invest. 28 (2019) (2019) 44–55.

[5] S. Savage, Lawful device access without mass surveillance risk: A technical design discussion, in: Proc. 25th ACM Conference on Computer and Communications Security, CCS, Toronto, Canada, 2018, pp. 1761–1774.

[6] D.-P. Le, H. Meng, L. Su, S.L. Yeo, V. Thing, BIFF: A blockchain-based IoT forensics framework with identity privacy, Proc. International Technical Conference of IEEE Region 10, TENCON, Jeju Island, Korea, 2018, pp. 2372–2377.

[7] Y. Zhang, S. Wu, B. Jin, J. Du, A blockchain-based process provenance for cloud forensics, in: Proc. 3rd IEEE International Conference on Computer and Communications, ICCC, Chengdu, China, 2017, pp. 2470–2473.

[8] A. Nieto, R. Rios, J. Lopez, Digital witness and privacy in IoT: Anonymous witnessing approach, in: Proc. 16th IEEE International Conference On Trust, Security And Privacy In Computing And Communications, TrustCom, Sydney, Australia, 2017, pp. 642–649.

[9] Z. Tian, M. Li, M. Qiu, Y. Sun, Shen Su, Block-DEF: A secure digital evidence framework using blockchain, Inform. Sci. 491 (2019) 151–165.

[10] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2009, https://bitcoin.org/bitcoin.pdf, Available: (Access date: 8 June 2020).

[11] A. Kosba, A.Miller, E. Shi, Z. Wen, C. Papamanthou, Hawk: The blockchain model of cryptography and privacy-preserving smart contracts, in: Proc. 37th IEEE Symposium on Security and Privacy (S & P), San Jose, USA, 2016, pp. 839–858.

[12] E. Androulaki, et al., Hyperledger fabric: A distributed operating system for permissioned blockchains, in: Proc. 13th European Conference on Computer Systems, EuroSys, Porto, Portugal, 2018, pp. 1–15.

[13] M. Li, L. Zhu, X. Lin, CoRide: A privacy-preserving collaborative-ride hailing service using blockchain-assisted vehicular fog computing, in: Proc. ACM 15th EAI International Conference on Security and Privacy in Communication Networks, SecureComm, 2019, Online.

[14] M. Conti, S. Kumar, C. Lal, S. Ruj, A survey on security and privacy issues of bitcoin, IEEE Commun. Surv. Tutor. 20 (4) (2018) 3416–3452.

[15] M. Easton, Met police 'corruption' claims lead to calls for investigation, 2016, https://www.bbc.com/news/uk-35394085, Available: (Access date: 8 June 2020).

[16] A. Nieto, R. Rios, J. Lopez, IoT-forensics meets privacy: Towards cooperative digital investigations, Sensors 18 (2) (2018) 492–506.

[17] J. Bowman, Pregnant woman killed in shooting was witness in high-profile Nathaniel Dixon murder trial, 2019, https://eu.citizen-times.com/story/news/2019/06/13/asheville-shooting-victim-witness-nathaniel-dixon-murder-trial-candace-pickens-ira-b-jones-park/1443558001, Available: (Access date: 8 June 2020).

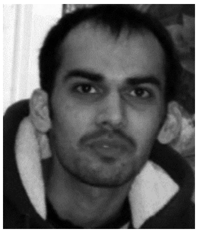
[18] S. Sharp, I. Plagianos, 'El Chapo' jurors could face a long-term threat: PTSD, 2019, https://www.latimes.com/nation/la-fg-new-york-el-chapo-juror-stress-20190204-story.html, Available: (Access date: 8 June 2020).

[19] R. Hartley-Parkinson, Two detectives jailed for sabotaging child abuse investigations, 2019, https://metro.co.uk/2019/05/10/two-detectives-sharon-patterson-lee-pollard-jailed-9494811, Available: (Access date: 8 June 2020).

[20] K. Yang, X. Jia, K. Ren, Attribute-based fine-grained access control with efficient revocation in cloud storage systems, in: Proc. 8th ACM Symposium on Information, Computer and Communications Security, ASIACCS, Hangzhou, China, 2013, pp. 523–528.

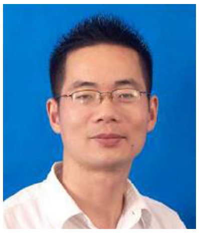
[21] M.R. Clarkson, S. Chong, A.C. Myers, Civitas: Toward a secure voting system, in: IEEE Symposium on Security and Privacy, S & P, Oakland, USA, 2008, pp. 354–368.
[22] Z.A. Baig, P. Szewczyk, C. Valli, P. Rabadia, P. Hannay, M. Chernyshev, M. Johnstone, P. Kerai, A. Ibrahim, K. Sansurooah, N. Syed, M. Peacock, Future challenges for smart cities: Cyber-security and digital forensics, Digit. Invest. 22 (2017) 3–13.
[23] J. Frankle, S. Park, D. Shaar, S. Goldwasser, D. Weitzner, Practical accountability of secret processes, in: Proc. 27th USENIX Security Symposium, USENIX Security, 2018, pp. 657–674.
[24] C. Grigoras, Applications of ENF criterion in forensic audio, video, computer and telecommunication analysis, Forens. Sci. Int. 167 (2–3) (2007) 136–145.
[25] G. France, Transparency of court proceedings. transparency international, 2019, https://knowledgehub.transparency.org/assets/uploads/helpdesk/Transparency-of-court-proceedings_2019_PR.pdf, Available: (Access date: 8 June 2020).
[26] Intel quickassist technology with intel key protection technology in intel server platforms based on intel xeon processor scalable family, 2020, https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/key-protection-technology-paper.pdf, Available: (Access date: 8 June 2020).
[27] C. Labrado, H. Thapliyal, Hardware security primitives for vehicles, IEEE Consum. Electron. Mag. 8 (6) (2019) 99–103.
[28] L. Thomson, Judicial ethics expert tells of misbehaving judges, 2020, https://www.deseret.com/2005/11/13/19922472/judicial-ethics-expert-tells-of-misbehaving-judges, (Access date: 12 August 2020).
[29] D. Pointcheval, O. Sanders, Short randomizable signatures, in: Proc. Cryptographers' Track at the RSA Conference, CT-RSA, San Francisco, USA, 2016, pp. 111–126.
[30] C.P. Schnorr, Efficient identification and signatures for smart cards, in: Proc. 9th Annual International Cryptology Conference, CRYPTO, Santa Barbara, USA, 1989, pp. 239–252.
[31] S. Hu, C. Cai, Q. Wang, C. Wang, X. Luo, K. Ren, Searching an encrypted cloud meets blockchain: A decentralized, reliable and fair realization, in: Proc. 37th IEEE Conference on Computer Communications, INFOCOM, Honolulu, USA, 2018, pp. 1–9.
[32] J. Benet, IPFS - content addressed, versioned, P2P file system (DRAFT 3), 2017, https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs.draft3.pdf, Available: (Access date: 8 June 2020).
[33] A. Antwi-Boasiako, H. Venter, Implementing the Harmonized Model for Digital Evidence Admissibility Assessment, in: Proc. IFIP International Conference on Digital Forensics, New Delhi, India, 2019, pp. 19–36.
[34] H. Corrigan-Gibbs, D. Kogan, Private Information Retrieval with Sublinear Online Time, in: Proc. 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, EUROCRYPT, Zagreb, Croatia, 2020, pp. 1–69.
[35] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Trans. Inform. Theory 31 (4) (1985) 469–472.
[36] Witness protection programmes EU experiences in the international context, 2013, http://www.europarl.europa.eu/RegData/bibliotheque/briefing/2013/130408/LDM_BRI(2013)130408_REV2_EN.pdf, Available: (Access date: 8 June 2020).
[37] Witness protection, 2019, http://www.rcmp-grc.gc.ca/en/witness-protection, Available: (Access date: 8 June 2020).
[38] Witness security program, 2019, https://www.usmarshals.gov/witsec, Available: (Access date: 8 June 2020).
[39] M. Altman, A. Wood, D.R. O'Brien, U. Gasser, Practical approaches to big data privacy over time, Int. Data Priv. Law 8 (1) (2018) 29–51.
[40] J.G. Chamani, D. Papadopoulos, C. Papamanthou, R. Jalili, New constructions for forward and backward private symmetric searchable encryption, in: Proc. 25th ACM Conference on Computer and Communications Security, CCS, Toronto, Canada, 2018, pp. 1038–1055.
[41] Thales, post-quantum crypto agility, 2020, https://safenet.gemalto.com/data-encryption/post-quantum-crypto-agility, Available: (Access date: 19 August 2020).
[42] IBM security architect advises post-quantum data protection, 2018, https://www.idquantique.com/ibm-security-architect-advises-post-quantum-data-protection, Available: (Access date: 8 June 2020).
[43] P. Ekparinya, V. Gramoli, G. Jourjon, The attack of the clones against Proof-of-Authority, 2020, https://arxiv.org/abs/1902.10244, Available: (Access date: 8 June 2020).
[44] S.S. Hasan, N.H. Sultan, F.A. Barbhuiya, Cloud data provenance using IPFS and blockchain technology, in: Proc. 17th International Workshop on Security in Cloud Computing, SCC, New York, USA, 2019, pp. 5–12.
[45] N. Nizamuddin, H. Hasan, K. Salah, IPFS-blockchain-based authenticity of online publications, in: Proc. 1s International Conference on Blockchain, ICBC, Seattle, USA, 2018, pp. 1–16.
[46] Hyperledger, 2020, https://www.hyperledger.org, Available: (Access date: 8 June 2020).
[47] A. Kiayias, Q. Tang, Traitor deterring schemes using Bitcoin as collateral for digital content, in: Proc. 22nd ACM Conference on Computer and Communications Security, CCS, Denver, USA, 2015, pp. 231–242.
[48] E.V. Mangipudi, K. Rao, J. Clark, A. Kate, Automated penalization of data breaches, 2018, pp. 1–12, Available: https://eprint.iacr.org/.

Meng Li (mengli@hfut.edu.cn) received the Ph.D. degree in Computer Science and Technology from Beijing Institute of Technology, Beijing, China in 2019, received the B.E. degree in the Information Security from Hefei University of Technology, Hefei, China, in 2010, and received the M.S. degree from Computer Science and Technology from Beijing Institute of Technology in 2013. He is now an Associate Researcher in the School of Computer Science and Information Engineering, Hefei University of Technology. He was sponsored by the China Scholarship Council to study as a visiting Ph.D. student in the Broadband Communications Research (BBCR) Lab at University of Waterloo and Wilfrid Laurier University from September 2017 to August 2018. His research interests include applied cryptography, security and privacy, vehicular networks, fog computing and blockchain.



Chhagan Lal received the bachelor's degree in computer science and engineering from MBM Engineering College, Jodhpur, India, in 2006, the master's degree in information technology with specialization in wireless communication from the Indian Institute of Information Technology, Allahabad, in 2009, and the Ph.D. degree in computer science and engineering from the Malaviya National Institute of Technology, Jaipur, India, in 2014. He is a Post-Doctoral Fellow with the Department of Mathematics, University of Padua, Italy. He has been awarded the Canadian Commonwealth Scholarship in 2012 under Canadian Commonwealth Scholarship Program to work in University of Saskatchewan, Saskatoon, SK, Canada. His current research areas include blockchain analysis, security in wireless networks, software defined networking, underwater acoustic networks, and context-based security solutions for Internet of Things networks.

Mauro Conti is Full Professor at the University of Padua, Italy, and Affiliate Professor at the University of Washington, Seattle, USA. He obtained his Ph.D. from Sapienza University of Rome, Italy, in 2009. After his Ph.D., he was a Post-Doc Researcher at Vrije Universiteit Amsterdam, The Netherlands. In 2011 he joined as Assistant Professor the University of Padua, where he became Associate Professor in 2015, and Full Professor in 2018. He has been Visiting Researcher at GMU, UCLA, UCI, TU Darmstadt, UF, and FIU. He has been awarded with a Marie Curie Fellowship (2012) by the European Commission, and with a Fellowship by the German DAAD (2013). His research is also funded by companies, including Cisco, Intel, and Huawei. His main research interest is in the area of Security and Privacy. In this area, he published more than 300 papers in topmost international peer-reviewed journals and conference. He is Area Editor-in-Chief for IEEE Communications Surveys & Tutorials, and Associate Editor for several journals, including IEEE Communications Surveys & Tutorials, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, and IEEE Transactions on Network and Service Management. He was Program Chair for TRUST 2015, ICISS 2016, WiSec 2017, ACNS 2020, and General Chair for SecureComm 2012 and ACM SACMAT 2013. He is Senior Member of the IEEE.



Donghui Hu (hudh@hfut.edu.cn) received the B.S. degree from Anhui Normal University, Wuhu, China, in 1995, the M.S. degree in computer science and technology from University of Science and Technology of China, Hefei, China, in 2004, and Ph.D. degree in information security from Wuhan University, Wuhan, China, 2010. He was a Visiting Researcher with UNCC IN 2013–2014. Currently, he is a Professor in the College of Computer Science and Information Engineering at the Hefei University of Technology. His research interests include information trustworthiness evaluation and privacy protection.