Anonymous, Secure, Traceable, and Efficient Decentralized Digital Forensics

Meng Li, *Senior Member, IEEE*, Yanzhe Shen, Guixin Ye, Jialing He, *Member, IEEE*, Xin Zheng, Zijian Zhang, *Senior Member, IEEE*, Liehuang Zhu, *Senior Member, IEEE*, and Mauro Conti, *Fellow, IEEE*

***Abstract*—Digital forensics is crucial to fight crimes around the world. Decentralized Digital Forensics (DDF) promotes it to another level by channeling the power of blockchain into digital investigations. In this work, we focus on the privacy and security of DDF. Our motivations arise from (1) how to track an anonymous-and-malicious data user who leaks only a part of the previously requested data, (2) how to achieve access control while protecting data from untrusted data centers, and (3) how to enable efficient and secure search on the blockchain. To address these issues, we propose Themis: an anonymous and secure DDF scheme with traceable anonymity, private access control, and efficient search. Our framework is boosted by establishing a Trusted Execution Environment in each authority (blockchain node) for securing the uploading, requesting, and searching. To instantiate the framework, we design a secure and robust watermarking scheme in conjunction with decentralized anonymous authentication, a private and fine-grained access control scheme, and an efficient and secure search scheme based on a dynamically updated data structure. We formally define and prove the privacy and security of Themis. We build a prototype with Ethereum and Intel SGX2 to evaluate its performance, which supports processing data from a considerable number of data providers and investigators.**

***Index Terms*—Access control, blockchain, decentralized digital forensics, efficiency, privacy, SGX2, security, watermark.**

Manuscript received 7 May 2023; revised 8 August 2023; accepted 30 September 2023. Date of publication 3 October 2023; date of current version 5 April 2024. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 62372149 and U23A20303, in part by the National Key Research and Development Program of China under Grant 2021YFB2701202, and in part by EU LOCARD Project under Grant H2020-SU-SEC-2018-832735. Recommended for acceptance by X. Yi. *(Corresponding authors: Zijian Zhang; Liehuang Zhu.)*

Meng Li is with the Key Laboratory of Knowledge Engineering with Big Data, Ministry of Education, School of Computer Science and Information Engineering, Anhui Province Key Laboratory of Industry Safety and Emergency Technology, Intelligent Interconnected Systems Laboratory of Anhui Province, Hefei University of Technology, Hefei, Anhui 230002, China (e-mail: mengli@hfut.edu.cn).

Yanzhe Shen is with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China (e-mail: shenyanzhe@hust.edu.cn).

Guixin Ye is with the School of Information Science and Technology, College of Computer Science, Northwest University, Xi'an, Shaanxi 710069, China (e-mail: gxye@nwu.edu.cn).

Jialing He is with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: hejialing@cqu.edu.cn).

Xin Zheng is with the Anhui Provincial Department of Justice, Wuhu, Anhui 241100, China (e-mail: xinzhenghf@gmail.com).

Zijian Zhang is with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100811, China, and also with the Southeast Institute of Information Technology, Beijing Institute of Technology, Beijing 351100, China (e-mail: zhangzijian@bit.edu.cn).

Liehuang Zhu is with the Department of Mathematics and HIT Center, University of Padua, 35131 Padua, Italy (e-mail: liehuangz@bit.edu.cn).

Mauro Conti is with the Department of Intelligent Systems, CyberSecurity Group, TU Delft, 2628 Delft, CD, Netherlands (e-mail: mauro.conti@unipd.it).

Digital Object Identifier 10.1109/TKDE.2023.3321712

I. Introduction

CRIMES have been raging all over the world. A report from the World Health Organization says that violence-related injuries kill 1.25 million people every year, which constitute 2.23% of all deaths [1]. To fight crimes, investigators resort to a variety of forensics expertise. However, traditional forensics methods suffer from low efficiency that delays the investigation progress. With the advancement and ubiquitousness of portable equipments, e.g., smartphones, electric vehicles, and drones, Digital Forensics (DF) enables investigators to timely gather and analyze evidence from data provides to carry out efficient investigations [2], [3], [4]. DF not only prevents victimization, but contributes to reducing violence and injustice, as well as building safe and resilient cities [5]. Therefore, it is counted as a key to fight against crimes in the information era.

Decentralized Digital Forensics (DDF), as we name it so for the first time, promotes the investigation process by invigorating DF with publicity, immutability, and verifiability due to the recent development of blockchain [6], [7]. As depicted in Fig. 1, a data provider collects data by using their equipment and sends captured data (potential evidence) to a blockchain; an investigator aperiodically requests data from the blockchain; after collaborating with a group of professionals on analyzing the data, the investigator files a report to the court to reach a verdict. DDF also facilitates complex investigations that span international borders and collaboration between different legal systems [4], [8].

Although DDF offers many benefits, it still encounters privacy and security issues [9], [10], e.g., user anonymity [11], data confidentiality [12], [13], unauthorized access [14], and intentional data leakage [15]. Some DDF schemes are proposed to solve the problems [15], [16], [17]. They build a forensics model and a threat model, design a protocol based on cryptographic primitives to protect privacy and security, and carefully integrate the protocol with the underlying blockchain network. Despite their targeted advantages, some issues still remain unresolved, which lead to our **three motivations** as follows.

M1: Decentralized Anonymous Authentication (DAA) with strong traceability. We notice a new attack called partial-data leakage attack where a malicious investigator leaks only a part of requested data to a blackmarket. Meanwhile, since we are constructing a digital forensics scheme based on blockchain,

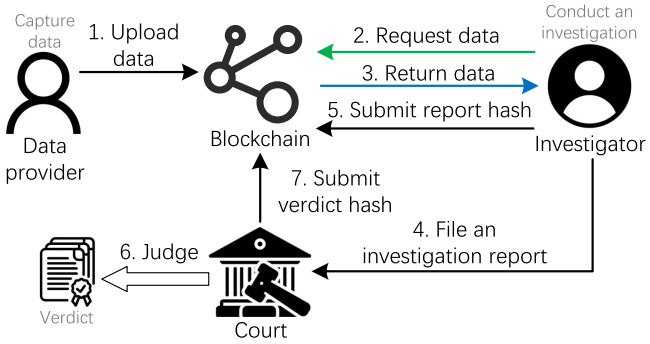


Fig. 1. Sketch model of DDF.

it is natural to rely on decentralized anonymous authentication [18] for user anonymity. However, among existing DAA schemes, Decentralized Anonymous Credential (DAC) [18] does not consider traceability and CanDID [19] does not mention how to reveal the real identity of a malicious user. Eunomia [15], a DDF scheme for vehicular networks, defends against data leakage attack, but it only works if the whole data is leaked.

M2: Access control secure against untrusted data centers. DDF can leverage Attribute-Based Encryption (ABE) to realize access control where data providers encrypts a message based on a set of attributes and a data user with relevant attributes can decrypt the ciphertext [20]. However, DDF data providers are not familiar with attributes. In fact, it is not necessary for them to know such information, thereby, there is no need for them to encrypt data with ABE. Meanwhile, authorities, i.e., data centers, may modify, add, or delete data intentionally or unintentionally [21], [22] due to various reasons, e.g., program glitches, security vulnerabilities, and misbehaving staff. Therefore, we consider the authorities to be untrusted. However, among recent decentralized access control schemes, Droplet's data providers have to know the public key of the data user they want to share their data with [23] and APECS's intermediary edge computing servers can still decipher the data [24]. Eunomia also exposes plaintext data to authorities.

M3: Efficient and secure query processing. It should be possible for investigators to request a data search by using predefined conditions, e.g., keywords and ranges. Such a search requires a secure index and a secure token for data privacy, but data users and investigators do not share keys, which is a requirement in searchable encryption and query processing schemes. Next, we notice that the authorities may return tampered or incomplete results for the same reasons above, thereby violating the expected security of the blockchain [26]. Therefore, result authenticity, i.e., correctness and completeness, is not guaranteed. Last, since the arrival time of data is uncertain, the authorities, as blockchain nodes, cannot generate an index that covers all the data quickly. Therefore, the index has to be efficiently updated. Among recent work of query on blockchain, GEM^2 Tree [25] and vChain [26] do not protect query privacy. Eunomia strenuously traverses all blocks to find the matched data. The aforementioned motivations and problems induce **four technical challenges** in styling anonymous, secure, and traceable DDF.

C1: From the anonymity vs. traceability perspective, how to preserve anonymity while defending against partial-data leakage attack in an anonymous environment? Data providers who value identity privacy will not contribute if anonymity is not guaranteed. Anonymous authentication reduces such concerns, but creates a breeding ground for partial-data leakage attacks from malicious investigators. Hence, it is challenging to trace traitors in an anonymous environment.

C2: From the access control vs. data privacy perspective, how to leverage ABE while keeping the data secure from the untrusted authorities? ABE-based access control grants data providers a great power of autonomy by encrypting the data with access structure and distributing secret keys to data users. In DDF, data providers cannot take such a responsibility and investigators (data users) do not communicate with all the data providers for a key. Even if we transform the encryption phase to the authorities as Eunomia did, we still suffer from potential data breach. Hence, it is challenging to securely perform ABE on untrusted authorities.

C3: From the efficiency vs. security perspective, how to conduct efficient and secure search on blockchain when there are enormous sources. In an open DDF system, the sheer amount of data will be uploaded in a temporally unpredictable manner. Such data puts forward a high requirement of efficient search on blockchain, i.e., the design of a data structure adapting to blockchain. Furthermore, we cannot sacrifice security for efficiency. Hence, it is challenging to search on blockchain efficiently and securely.

C4: From a formalization perspective, how to formally define and prove privacy and security for DDF? Our target DDF scheme is basically a security protocol including parties, functions, attacks, and goals. Different from standard cryptographic protocols, it needs careful adjustments when capturing privacy and security in DDF.

To tackle these challenges, we build a framework based on Trusted Execution Environment (TEE) [27], [28] and consortium blockchain [29]. We leverage three carefully bridged primitives to propose a novel DDF scheme Themis. It provides traceable anonymity, private access control, and efficient search. Specifically, data providers register with authorities (blockchain nodes) to obtain a DAC by using a threshold signature scheme [30], [31]. They upload encrypted data to an authority equipped with an enclave. The encalve decrypts the ciphertext and produces (updates) a secure index [32] to generate a dynamically updated index tree [25]. Only the root hash instead of the entire tree is stored in a smart contract. Investigators send a data request to an authority that searches the tree with the request. If there is a match, the encalve verifies key, embeds the key into matched data [33] by using an improved watermarking scheme, and then performs ABE [20]. We frame three key contributions as follows.

- We put forth the concept of DDF and propose a security framework based on TEE and blockchain. Specifically, we define and defend against the partial-data leakage attack,

assume the authorities to be untrusted, and search on blockchain efficiently and securely.
- We design a secure and robust watermarking scheme in conjunction with decentralized anonymous authentication, a private and fine-grained access control scheme, and an efficient and secure search scheme with a dynamically updated data structure.
- We formally define and prove the privacy and security. We build a prototype based on the pioneering Ethereum and SGX2. We conduct extensive experiments to evaluate performance while comparing with existing work.

The remainder of this paper is organized as follows. We review some related work in Section II. Section III introduces some preliminaries. Section IV formalizes the problem. In Section V, we elaborate Themis, followed by the privacy and security analysis in Section VI and performance evaluation in Section VII, respectively. Finally, we provide some discussions in Section VII and conclude the paper in Section IX.

TABLE I
KEY NOTATIONS OF THEMIS

Notation	Definition
DF, DDF	Digital Forensics, Decentralized Digital Forensics
DAA	Decentralized Anonymous Authentication
DAC, cre	Decentralized Anonymous Credential
ABE	Attribute-Based Encryption
TEE	Trusted Execution Environment
SGX	Software Guard eXtensions
PPT	Probabilistic Polynomial-Time
DP, IN	Data Provider, Investigator
AU, EV, CB	Authority, enclave, consortium blockchain
λ, key	Security parameter, secret key
$\mathbb{A}$, $\mathcal{S}$, sk	Access structure, attribute set, secret key
$\mathcal{SK}$, k	Secret key set, secret key
$\{h_1, \cdots, h_v\}$, h	pseudo-random hash function, hash function
n_0, n_1, n_2	Number of AUs, number of DPs, number of INs
d, wd, pwd	Data, watermarked data, a part of wd
ed, dm, hv	Encrypted data, data message, hash value
tk, kp, dr	Token, key proof, data request
$\mathcal{B}$, L	Indistinguishable Bloom filter, length of $\mathcal{B}$
rr, er,	Request result, encrypted request result
ev, puk, prk	Enclave, public key, private key
$\mathrm{Tx}_2^{\mathrm{Upl}}$, $\mathrm{Tx}_3^{\mathrm{Upd}}$,	Data uploading (root updating) transaction
$\mathrm{Tx}_3^{\mathrm{ReqD}}/\mathrm{Tx}_3^{\mathrm{ReqG}}$	Data request deny/grant transaction
$\mathcal{A}$, $\mathcal{C}$, Π	Adversary, challenger, Themis scheme

II. RELATED WORK

A. DF

A typical DF process consists of several key steps, such as evidence acquisition and duplication, evidence analysis, and result presentation [2]. An investigator collects data from crime scenes and witnesses by using professional acquisition tools, and send the data to an analyst for systematical analysis to look for potential evidence. During the analysis, different techniques including automatic detection image recognition, and image reconstruction, are used to process a large number of digital files. Based on an analysis report and proper crime reasoning, the investigator produces an investigation report.

Garfinkel [3] stated that we were in the "Golden Age of Digital Forensics" (1999-2007) when DF could magically see into the past and into the criminal mind. But is was coming to an end. This is because the collected data come in different formats and ciphertexts, and they cannot be analyzed with tools of the time. Next, the author proposes a plan for an improvement in research through the adoption of systematic approaches for representing forensic data and performing forensic computation.

Caviglione et al. [4] pointed out that developments in DF forensics were in a difficult situation since its evolution is seriously challenged by the increasing popularity of digital devices and the heterogeneity of the hardware and software platforms. The majority of forensic software is not applicable to identify anomalies in an unattended way. Therefore, one of the major challenges for DF is the design of tools and techniques to process the data and report possible evidences to investigators for further investigation.

B. DDF

Cebe et al. [16] stated that connected and smart vehicles would provide valuable data to stakeholders, e.g., maintenance companies, vehicle manufacturers, drivers, and insurance companies. They designed a framework Block4Forensic for managing vehicular data. The framework combined a public key infrastructure (i.e., pseudonym certificates) with a permissioned blockchain to establish privacy-preserving membership. They proposed a fragmented ledger to preserve diagnosis records and maintenance reports. However, the authors just propose a framework design but do not give a concrete implementation.

Li et al. [17] proposed a blockchain-based vehicular DF scheme BB-VDF consisting of accountable protocols and privacy-preserving techniques. They modeled the warrant state as a finite state machine and realized the state transition via a smart contract. They used distributed key generation in a threshold cryptosystem to require an investigator to have at least t decryption shares to decrypt the requested data. They also designed a distributed key-policy ABE scheme to realize the access control. However, BB-VDF scheme does not address anonymity or traceability.

Li et al. [15] presented a blockchain-based vehicular DF scheme Eunomia with anonymity, access control, and traceability. Different from BB-VDF, they modeled a forensics procedure as a finite state machine. They leverage decentralized anonymous credentials [21] to provide user anonymity. They adopted a ciphertext-policy ABE [20] to realize access control. They defended against data leakage by executing an interactive protocol between an authority and a investigator to securely watermark the investigator's secret key into the requested data [34]. However, Eunomia cannot defend the partial-data leakage attack or provide result authenticity.

Themis's promotion over existing work is threefold. First, Themis provides traceable anonymity where honest data providers and investigators remain anonymous but malicious investigators who leaks (even just a part of) data will be tracked. Existing work [15] can only trace the tractor when the whole data is leaked. Second, Themis achieves private access control where authorities control investigators' access to data but cannot access the plaintext data directly. Third, Themis enables efficient and

TABLE II
COMPARISON ON MODEL, PRIVACY, SECURITY, AND FUNCTIONALITY WITH EXISTING WORK

Property	Decentralized	Anonymity	Unlinkability	Private Access Control	Authentication	Strong Traceability
Block4Forensic [16]	Partial[a]				√	
BB-VDF [17]	Partial[a]			Partial[b]	√	
Eunomia [15]	Partial[a]	√	√	Partial[b]	√	Partial[c]
Themis	√	√	√	√	√	√

a: There is a centralized and trusted third party. *b*: It only provides access control. *c*: It can only trace a traitor who leaks all the requested data.

secure search on blockchain where the index of uploaded data is being efficiently and securely updated on both the authorities and the smart contract. We compare Themis with existing work in Table II in Section VI.

III. PRELIMINARIES

A. Threshold Signature

A threshold signature scheme is constructed upon a signature scheme [30] and a distributed key generation protocol [31]. It consists of four algorithms as follows:

$\mathsf{Gen}(1^{\lambda_1})$: given a security parameter λ_1, outputs a private key prk, n shares of prk denoted as $\{prk_1, prk_2, \ldots, prk_{n_0}\}$, and a public key puk.

$\mathsf{Sign}(prk_i, m)$: given a private key share prk_i and a message m, outputs a signature σ_i.

$\mathsf{Comb}(\sigma_1, \sigma_2, \ldots, \sigma_t)$: given t partial signatures σ_1, σ_2, $\cdots$, σ_t, outputs a combined signature σ.

$\mathsf{Verify}(puk, m, \sigma)$: given a public key puk, a message m, and a signature σ, outputs 1 if σ is a valid signature on m, and 0 otherwise.

B. Ciphertext-Policy Attribute-Based Encryption

A Ciphertext-Policy Attribute-Based Encryption (CP-ABE) scheme [35] consists of four algorithms as follows:

$\mathsf{Setup}(1^{\lambda_2})$: given a security parameter λ_2, outputs a public key mpk and a master secret key msk.

$\mathsf{Encrypt}(mpk, \mathbb{A}, m)$: given the public key mpk, an access structure $\mathbb{A}$, and a message m, outputs a ciphertext ct.

$\mathsf{KeyGen}(msk, \mathcal{S})$: given the master secret key msk and a set of attributes $\mathcal{S}$, outputs a secret key sk.

$\mathsf{Decrypt}(mpk, ct, sk)$: given the public key mpk, a ciphertext ct, and a secret key sk, outputs a message m in the message space $\mathcal{M}$.

C. Digital Watermarking

A robust digital watermarking scheme [33] consists of three algorithms:

$\mathsf{KGen}(1^{\lambda_3})$: given a security parameter λ_3, outputs a secret key key.

$\mathsf{Embed}(key, m)$: given a secret key key, a message m, generates a watermark wm, embeds wm into m, outputs a watermarked message m'.

$\mathsf{Extract}(key, m')$: given a secret key key and a watermarked message m', outputs a watermark wm.

D. Privacy-Preserving Query Processing

A privacy-preserving query processing scheme consists of five algorithms.

$\mathsf{Initialize}(1^{\lambda_4})$: given a security parameter λ_4, outputs an encryption key k_0, secret keys $\mathcal{SK} = \{k_1, k_2, \ldots, k_{v+1}\}$, $v+1$ pseudo-random hash functions $\mathcal{H} = \{h_1, h_2, \ldots, h_{v+1}\}$, a hash function h.

$\mathsf{Enc}(k_0, di)$: given the secret key k_0 and a data item di, outputs a ciphertext ci.

$\mathsf{IndexGen}(\mathcal{SK}, \mathcal{H} = \{h_1, \ldots, h_v, h_{v+1}\}, h, m)$: given the secret keys $\mathcal{SK}$, the set of pseudo-random hash functions $\mathcal{H}$, the hash function h, and an data item di, outputs a secure index $\mathcal{B}$. Specifically, the index $\mathcal{B}$ has L twins and each twin stores either 0 or 1. $\mathcal{H}$ determines which cell stores 1. di is hashed to L twin cells $\mathcal{B}[h(di)][h(h_{v+1}(h_i(di)) \oplus r_{\mathcal{B}})] = 1$ where $i \in [1, v]$ and $r_{\mathcal{B}}$ is a random number.

$\mathsf{TokenGen}(\mathcal{SK}, \mathcal{H}, h, q)$: given the secret key set $\mathcal{SK}$, the pseudo-random hash functions $\mathcal{H}$, the hash function h, and a query q, outputs a token tk.

$\mathsf{Search}(\mathcal{B}, tk)$: given a secure index $\mathcal{B}$ and a token tk, outputs a bit. Specifically, the tk is hashed into $\mathcal{B}$ to check if it exists in $\mathcal{B}$: $\mathcal{B}[h(tk)][h(h_{v+1}(h_i(tk)) \oplus r)] = 1$ for $i \in [i, v]$.

$\mathsf{Dec}(sk, ci)$: given a secret key sk and a ciphertext ci, outputs a data item di.

E. Intel SGX2

Software Guard eXtensions (SGX) is a hardware extension of Intel Architecture that enables an application to establish a protected execution space, i.e., an enclave [36], [37], [38]. SGX stores enclave pages and SGX structures in the protected memory called Enclave Page Cache (EPC). SGX guarantees confidentiality of code/data and detection of an integrity violation of an enclave instance from software attacks [36]. SGX allows one to verify that a piece of software has been correctly instantiated on the platform via attestation.

With SGX, developers can build trusted modules inside an application to protect secrets. Some works have been devoted to developing a security framework based on SGX. For instance, PrivacyGuard [39] leverages smart contracts and TEE to help data owners' control over the access and usage of their private data. SecGrid [40] leverages SGX to ensure that grid utilities efficiently perform rich functionalities on users' private data while guaranteeing their privacy.

Since SGX imposes limitations regarding memory commitment and reuse of enclave memory, Intel introduces SGX2 to extend the SGX instruction set to include dynamic memory management support for enclaves [27], [28]. SGX2 instructions offer software with more capability to manage memory and page

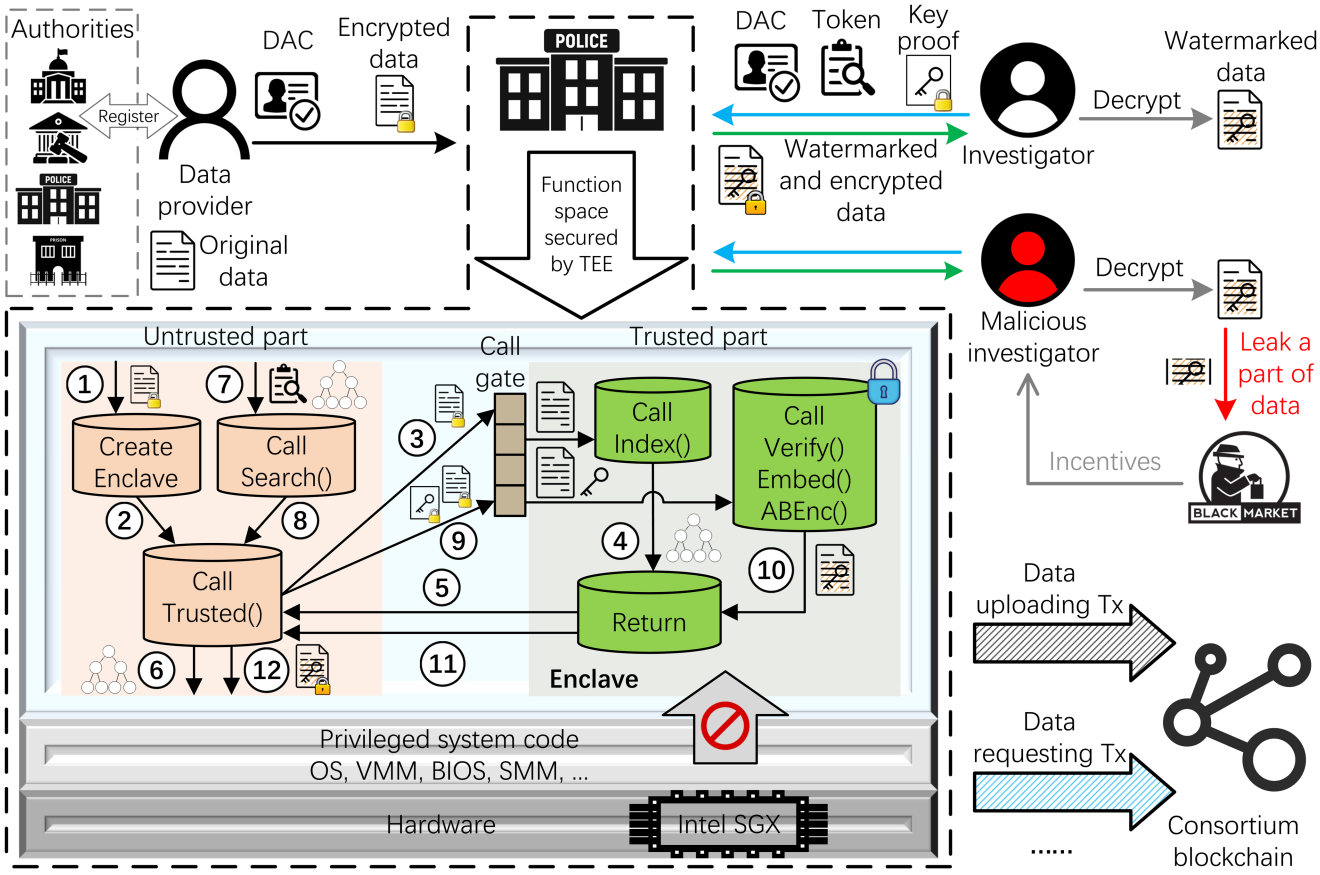


Fig. 2. Architectural vision of Themis.

protections from inside an enclave while preserving the security of the SGX architecture and system software.

F. Consortium Blockchain

As an underlying technique in Bitcoin [6], blockchain is a public ledger recording transactions among untrustworthy users in a decentralized network. The transactions are packed into separate blocks by a set of nodes using a predefined consensus algorithm, and the blocks are sequentially linked into a chain by their cryptographic hashes. Consortium blockchain is a specific blockchain maintained by a group of authorized entities. Only qualified parties are allowed to access the blockchain. It aims to secure transactions between users who do not fully trust each other but work collaboratively toward a common goal. Its consensus process is controlled by the authorized entities. Some noted consortium blockchains are Hyperledger Fabric, EEA, and Corda.

IV. Problem Statement

A. System Model

The system model of Themis consists of four types of entities: Data Provider, Investigator, Authority, and Consortium Blockchain. We display the architectural vision of Themis in Fig. 2 and list the key notations in Table I.

Data Provider (DP) is a person who holds a device with sensing capabilities. She/he uploads to an authority the data that is captured by the device to provide potential evidence. Such an act is beneficial to uphold justice, which is rewarded by the police. DP can be a device that is installed in a fixed place or cruising along a route, e.g., a cctv camera and autonomous vehicle. A DP first registers to a group of authorities to compute a DAC cre for anonymity. Next, the DP encrypts data d into ed, uploads a data message $dm = (cre, ed)$ and a hash value hv of ed to an authority.

Investigator (IN) is a professional who performs an investigation task to solve a civil/criminal case. She/he registers to obtain a DAC cre and registers to their department for a set of attributes $\mathcal{S}$, a secret key sk, and a secret key set $\mathcal{SK}$. The IN computes a token tk and a key proof kp, and sends a data request $dr = (cre, tk, kp)$, and a hv of dr to an authority. After receiving an encrypted request result er from the authority, the IN decrypts er to obtain request result rr. The IN cooperates with other professional and continues to request data from the blockchain if necessary. After the case is solved, the IN sends a reporting transaction Tx^{Rep} to the blockchain.

Authority (AU) is an institute related to digital forensics, such as police department, transportation department, hospital, legislative council, prison, and court. Each authority acts as a blockchain node to maintain a consortium blockchain. When receiving a data message, an AU authenticates its DP, computes

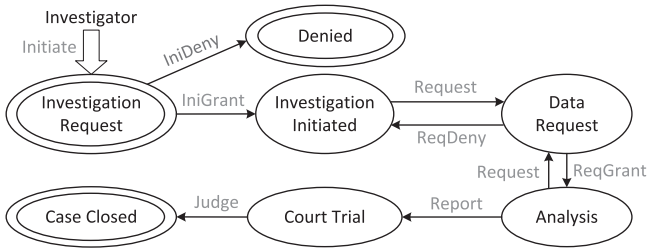


Fig. 3. State machine model of DDF.

a secure index, stores the encrypted data from DPs in a local database, and broadcasts a data uploading transaction $\mathrm{Tx}^{\mathrm{Upl}}$ to the blockchain. After receiving a data request, an AU authenticates its IN, searches the database, verifies the IN's key, embeds the key into matched data, encrypts the watermarked data, returns an encrypted request result er to the IN, and broadcasts a data requesting transaction $\mathrm{Tx}^{\mathrm{Req}}$. After a case is closed and a trial is concluded, the court broadcast a judging transaction $\mathrm{Tx}^{\mathrm{Jud}}$. Specifically, each AU is initiated with a TEE, i.e., SGX2 enclave (EV), to secure the index building, key verification, key embedding, and encryption, such that they cannot acquire the plaintext of uploaded data.

Consortium Blockchain (CB) is a blockchain that is maintained by authorities. The CB records the data transactions, such as $\mathrm{Tx}^{\mathrm{IniG}}$, $\mathrm{Tx}^{\mathrm{IniD}}$, $\mathrm{Tx}^{\mathrm{Upl}}$, $\mathrm{Tx}^{\mathrm{ReqG}}$, $\mathrm{Tx}^{\mathrm{ReqD}}$, $\mathrm{Tx}^{\mathrm{Rep}}$, and $\mathrm{Tx}^{\mathrm{Jud}}$. We model DDF as a finite state machine and record state transitions in a smart contract. There are seven states as depicted in Fig. 3.

We choose a consortium blockchain to lay the foundation of the whole system because the consortium blockchain is a permissioned blockchain that only allows qualified parties to maintain the blockchain. Therefore, it suits digital forensics where several authorities who do not fully trust each other need to collaborate to manage forensics data while being incompatible with the centralization of private blockchains or arbitrary participation of public blockchains.

B. Threat Model

We adopt the classic honest-but-curious assumption for most internal entities. A part of INs and AUs may act maliciously. A malicious IN launches three attacks. **Partial-data leakage attack**: leaking a part of data, i.e., a corrupted investigator crops a picture and leaks the cropped picture to a blackmarket. A malicious AU can launch a data tampering attack and a data ignoring attack. **Tampering attack**: tampering with the returned data that is stored in the local dataset. **Ignoring attack**: ignoring some stored data during data search.

Definition 1 (Attacks): Given a dataset $\{d_1, d_2, \ldots, d_{n_1}\}$, a watermarked dataset $\{wd_1, wd_2, \ldots, wd_{n_1}\}$, two secret keys sk, key, a secure index $\mathcal{T}$, a data request dr, an ABE public key pk, and a malicious IN with a data request dr and a secret key sk, partial-data leakage attack Att^{L}, tampering attack Att^{T}, and ignoring attack Att^{I} are defined:

$$Att^{\mathrm{L}} : \mathcal{T} \times \{dr\} \to \{wd_j | \mathsf{Search}(\mathcal{T}, dr) = \{c_i\},$$
$$\mathsf{Dec}(pk, c_i, sk) = \mathsf{Embed}(key, d_i), \exists wd_i : \ wd_j \subset wd_i\},$$
$$Att^{\mathrm{T}} : \mathcal{T} \times \{dr\} \to \{wd_j | \mathsf{Search}(\mathcal{T}, dr) = \{c_i\},$$
$$\mathsf{Dec}(pk, c_i, sk) = \mathsf{Embed}(key, d_i), \nexists wd_i : wd_j = wd_i\},$$
$$Att^{\mathrm{I}} : \mathcal{T} \times \{dr\} \to \{wd_j | \mathsf{Search}(\mathcal{T}, dr) = \{c_i\},$$
$$\mathsf{Dec}(pk, c_i, sk) = \mathsf{Embed}(key, d_i), \{wd_j\} \subset \{wd_i\}\}.$$

Given a secure index $\mathcal{T}$ and a data request dr, the search result is $\mathsf{Search}(\mathcal{T}, dr) = \{c_i\}$ and the decryption of each c_i belongs to the original dataset: the partial-data leakage attack Att^L returns a set of cropped watermarked data $\{wd_j\}$ where each wd_j is a part of the original watermarked data wd_i, the tampering attack Att^T returns a set of tampered watermarked data $\{wd_j\}$ where every wd_j does not belong to the original watermarked dataset, and the ignoring attack Att^I returns a set of watermarked data $\{wd_j\}$ which is a subset of the original watermarked dataset.

We also consider a **collusion attack** between AU and IN, where the two colluding parties share information to disclose the identity and data of a DP.

C. Design Objectives

We provide formal security definitions by using cryptographic experiments (games) [41]

Privacy is twofold. (1.1) **Anonymity**: the real identity of DPs/AUs is kept anonymous against $\mathcal{A}$ when they are uploading/requesting data to/from the CB. Assume $\mathcal{DP} = \{dp_1, dp_2, \cdots, dp_{n_1}\}$ is a set of DPs' pseudo identities, $\mathcal{IN} = \{in_1, in_2, \cdots, in_{n_2}\}$ is a set of INs, $\mathcal{A}$ is Probabilistic Polynomial-Time (PPT) adversary, $\mathcal{C}$ is a challenger, Π is the proposed Themis scheme, and λ_0 is a security parameter. We define a *DP-anonymity experiment* $\mathsf{Exp}^{\text{Any-DP}}_{\mathcal{A},\Pi}(\lambda_0)$ as follows:

1. Setup: $\mathcal{C}$ prepares $\mathcal{DP}$ and $\mathcal{IN}$.
2. Execution: $\mathcal{C}$ executes Π with $\mathcal{DP}$ and $\mathcal{IN}$.
3. Challenge: $\mathcal{C}$ chooses a random $k \in \{1, 2, \cdots, n_1\}$ and sends dp_k to $\mathcal{A}$.
4. Guess: $\mathcal{A}$ outputs a value k'. $\mathcal{A}$ wins if $k' = k$.

We compute a DP-anonymity advantage of $\mathcal{A}$ in correctly guessing k in $\mathsf{Exp}^{\text{Any-DP}}_{\mathcal{A},\Pi}(\lambda_0)$. We denote it by

$$\mathsf{Adv}^{\text{Any-DP}}_{\mathcal{A},\Pi}(\lambda_0) = \left| \Pr[k' = k] - \frac{1}{n_1} \right|.$$

Similarly, the experiment $\mathsf{Exp}^{\text{Any-IN}}_{\mathcal{A},\Pi}(\lambda_0)$ is defined where the target set is $\mathcal{IN}$, $\mathcal{C}$ chooses k from $[1, n_2]$ and $\mathcal{A}$ outputs $k' \in [1, n_2]$. $\mathsf{Adv}^{\text{Any-IN}}_{\mathcal{A},\Pi}(\lambda_0) = |\Pr[k' = k] - \frac{1}{n_2}|$ under no collusion attack; and $|\Pr[k' = k] - \frac{1}{n_2 - 1}|$ otherwise.

Definition 2 (Anonymity): Π achieves anonymity if

$$\mathsf{Adv}^{\text{Any-DP}}_{\mathcal{A},\Pi}(\lambda_0) \le \mathsf{negl}(\lambda_0), \ \mathsf{Adv}^{\text{Any-IN}}_{\mathcal{A},\Pi}(\lambda_0) \le \mathsf{negl}(\lambda_0).$$

(1.2) **Unlinkability**: two data uploading/requesting transactions sent by the same DP/IN are indistinguishable to $\mathcal{A}$. We define a *DP unlinkability experiment* $\mathsf{Exp}^{\text{Unl-DP}}_{\mathcal{A},\Pi}(\lambda_0, \lambda_1)$ as follows:

1. Setup: $\mathcal{C}$ prepares $\mathcal{DP}$ and $\mathcal{IN}$.
2. Execution: $\mathcal{C}$ executes Π with $\mathcal{DP}$ and $\mathcal{IN}$. $\mathcal{A}$ generates two data d_0, d_1, and sends them to $\mathcal{C}$.

3. Challenge: $\mathcal{C}$ chooses a random $k \in \{1, 2, \ldots, n_1\}$, prepares $\{dm_1, dm_2, \ldots, dm_{n_1}\}$, dm'_k, and sends them to $\mathcal{A}$. dm_k and dm'_k are computed by the same DP.
4. Guess: $\mathcal{A}$ outputs a value k'. $\mathcal{A}$ wins if $k' = k$.

An Experiment $\mathsf{Exp}^{\text{Unl-IN}}_{\mathcal{A},\Pi}(\lambda_0, \lambda_1)$ is defined regarding $\{dr_1, dr_2, \ldots, dr_{n_2}\}$ and dr'_k. We define $\mathsf{Adv}^{\text{Unl-DP}}_{\mathcal{A},\Pi}(\lambda_0, \lambda_1) = |\Pr[k' = k] - \frac{1}{n_1}|$. $\mathsf{Adv}^{\text{Unl-IN}}_{\mathcal{A},\Pi}(\lambda_0, \lambda_1) = |\Pr[k' = k] - \frac{1}{n_2}|$ under no collusion attack; and $|\Pr[k' = k] - \frac{1}{n_2 - 1}|$ otherwise.

Definition 3 (Unlinkability): Π achieves unlinkability if

$$\mathsf{Adv}^{\text{Unl-DP}}_{\mathcal{A},\Pi}(\lambda_0, \lambda_1) \leq \mathsf{negl}(\lambda_0) + \mathsf{negl}(\lambda_1),$$

$$\mathsf{Adv}^{\text{Unl-IN}}_{\mathcal{A},\Pi}(\lambda_0, \lambda_1) \leq \mathsf{negl}(\lambda_0) + \mathsf{negl}(\lambda_1).$$

Security contains two objectives. (2.1) **Private access control**. The plaintext data is protected from AUs. The accesses to data from unqualified INs are denied. We define an *AU-private access control experiment* $\mathsf{Exp}^{\text{PAC-AU}}_{\mathcal{A},\Pi}(\lambda_0, \lambda_2)$ as follows:

1. Setup: $\mathcal{C}$ prepares $\mathcal{DP}$ and $\mathcal{IN}$.
2. Execution: $\mathcal{C}$ executes Π with $\mathcal{DP}$ and $\mathcal{IN}$. $\mathcal{A}$ generates two data d_0, d_1, and sends them to $\mathcal{C}$.
3. Challenge: $\mathcal{C}$ chooses a random bit $b \in \{0, 1\}$, computes a ciphertext c_b of d_b, and sends c_b to $\mathcal{A}$.
4. Guess: $\mathcal{A}$ outputs a bit b'. $\mathcal{A}$ wins if $b' = b$.

We define $\mathsf{Exp}^{\text{PAC-IN}}_{\mathcal{A},\Pi}(\lambda_0, \lambda_2)$ as follows:

1. Setup: $\mathcal{C}$ prepares $\mathcal{DP}$ and $\mathcal{IN}$. $\mathcal{A}$ sends $\mathcal{S}_{\mathcal{A}}$ to $\mathcal{C}$ who returns $sk_{\mathcal{A}} \leftarrow \mathsf{KeyGen}(msk, \mathcal{S}_{\mathcal{A}})$ to $\mathcal{A}$.
2. Execution: $\mathcal{C}$ executes Π with $\mathcal{DP}$ and $\mathcal{IN}$. $\mathcal{A}$ generates two data values d_0, d_1, and sends them to $\mathcal{C}$.
3. Challenge: $\mathcal{C}$ chooses a random bit $b \in \{0, 1\}$, chooses m_b for some dm, outputs its ciphertext c_b for its er, and sends c_b to $\mathcal{A}$.
4. Guess: $\mathcal{A}$ outputs a bit b'. $\mathcal{A}$ wins if $b' = b$.

We define $\mathsf{Adv}^{\text{PAC-AU}}_{\mathcal{A},\Pi}(\lambda_0, \lambda_2) = |\Pr[k' = k] - \frac{1}{2}|$ and $\mathsf{Adv}^{\text{PAC-IN}}_{\mathcal{A},\Pi}(\lambda_0, \lambda_2) = |\Pr[k' = k] - \frac{1}{2}|$.

Definition 4 (Private Access Control): Π achieves private access control if

$$\mathsf{Adv}^{\text{PAC-AU}}_{\mathcal{A},\Pi}(\lambda_0, \lambda_2) \leq \mathsf{negl}(\lambda_0, \lambda_2),$$

$$\mathsf{Adv}^{\text{PAC-IN}}_{\mathcal{A},\Pi}(\lambda_0, \lambda_2) \leq \mathsf{negl}(\lambda_0, \lambda_2).$$

(2.2) **Authentication**. The identity of DPs and INs should be authenticated. We define an *authentication experiment* $\mathsf{Exp}^{\text{Aut}}_{\mathcal{A},\Pi}(\lambda_1)$ as follows:

1. Setup: $\mathcal{C}$ prepares $\mathcal{DP}$ and $\mathcal{IN}$.
2. Execution: $\mathcal{C}$ executes Π with $\mathcal{DP}$ and $\mathcal{IN}$. $\mathcal{A}$ is given 1^{λ_1} and oracle access to $\mathsf{Sign}(prk_i, \cdot)$. $\mathcal{A}$ can only submit queries to at most w signing oracles.
3. Guess: $\mathcal{A}$ outputs (m, σ). $\mathcal{A}$ wins if $m \neq \mathcal{Q}$ and $\mathsf{Verify}(pk, m, \sigma) = 1$, denoted by $\mathsf{Exp}^{\text{Aut}}_{\mathcal{A},\Pi}(\lambda_1) = 1$.

Definition 5 (Authentication): Π achieves authentication if

$$\Pr[\mathsf{Exp}^{\text{Aut}}_{\mathcal{A},\Pi}(\lambda_1) = 1] \leq \mathsf{negl}(\lambda_1).$$

Efficiency refers to efficient performance, i.e., the computational costs, communication overhead, and gas costs (in Ethereum) should be as low as possible.

Function refers to strong traceability where Themis can track the IN who leaks a part of watermarked data pwd.

V. Proposed Scheme

A. Overview

There are eight phases in Themis, namely system initialization, entity registration, data uploading, data requesting, analyzing, reporting, trailing, and traitor tracing. We use an end-to-end example to show the general process of DDF. A group of local administrative institutes, as AUs, establish a CB. Alice, as a DP, witnessed a bank robbery and took two pictures of a suspect with her smartphone. Given a Themis app on her smartphone, Alice registers to AUs to acquire a DAC. Then, Alice uploads the two pictures to the CB by sending two encrypted photos and the DAC to a local police department, i.e., an AU. The local police department verifies the DAC, decrypts the encrypted data into the enclave, computes a secure index, encrypts the data, and stores the ciphertext in its local database. If more data arrived later, an index tree is built and updated. A crime investigator Bob, as a DU, initiates an investigation on the robbery. He sends a DAC, a token, and a key proof to the police department. The police department verifies the DAC and the key proof, searches corresponding data, watermarks Bob's key into the matching data within an enclave, and sends watermarked-and-encrypted data back to the CPI. As for analyzing, reporting, and trailing, we follow the idea of [16] and [15]. If Bob leaks requested data, the authority can trace back to Bob by extracting his key from the leaked data.

B. System Initialization

Before any security protocols are preset, a secure enclave EV ev_i is installed at each AU au_i ($1 \leq i \leq n_0$). A pair of private key and public key (prk_{ev_i}, puk_{ev_i}) is generated for ev_i by SGX Seal method. $\Omega = (\mathsf{S}, \mathsf{E}, \mathsf{D})$ is an asymmetric encryption scheme with a security parameter λ_0.

Given a security parameter λ_1, a distributed key generation protocol [31] to generate a secret key $prk = d$, a public key $pk = (n = pq, e)$, and a polynomial $f(x) = \sum_{i=0}^{t-1} a_i x^i$, where e is a prime bigger than n_0, $de \equiv 1 \bmod M$, $a_0 = d$, and a_i ($1 \leq i \leq t-1$) is random chosen from $\{0, 1, \ldots, M-1\}$. At the end, n_0 secret shares of prk are computed as $prk_i^{ev} = f(i) \bmod M$, $1 \leq i \leq n_0$, a random value is chosen as $v \in Q_n$, and n_0 verification keys are computed as $v_i = v^{s_i}$, $1 \leq i \leq n_0$ [30]. Given a security parameter λ_2, $\mathsf{Setup}(1^{\lambda_2})$ outputs a public key mpk and a master secret key msk. Given a security parameter λ_3, $\mathsf{KGen}(1^{\lambda_3})$ outputs a secret key key. Given a security parameter λ_4, $\mathsf{Initialize}(1^{\lambda_4})$ outputs secret keys $\mathcal{SK} = \{k_1, k_2, \ldots, k_{v+1}\}$, $v+1$ pseudo-random hash functions $\mathcal{H} = \{h_1, h_2, \ldots, h_{v+1}\}$, and a hash function h. Here, $h_i = \mathsf{HMAC}(k_i, \cdot)\%L (1 \leq i \leq v)$ where L is the length of the underlying indistinguishable Bloom filter $\mathcal{B}$, $h_{v+1} = \mathsf{HMAC}(k_{v+1}, \cdot)$, $h = \mathsf{SHA256}(\cdot)\%2$.

All AUs determine the parameters (e.g., consensus algorithm, block generation time, node account deposit) for the CB. An Investigation Smart Contract (ISC) with a unique address $\mathrm{Add}_{\mathrm{ISC}}$ is published.

Finally, msk, key, and $\mathcal{SK}$ are securely distributed to all enclaves. Public parameters $pp = (\{puk_{ev_i}\}_{i=1}^{n_0}, v, \{v_i\}_{i=1}^{n_0}, pk, mpk, \mathcal{H}, h)$ are published.

C. Entity Registration

A DP generates a pair of private key and public key (prk_{dp}, puk_{dp}) and registers to a subset of t EVs $\mathcal{EV} = \{ev_{i_1}, ev_{i_2}, \ldots, ev_{i_t}\}$ for a DAC cre_{dp} as follows:

- DP generates a random registration message rm and submits $erm = \mathsf{E}(puk_{ev_{i_j}}, "\text{Register}"||rm)$ to ev_{i_j}.
- Each ev_{i_j} decrypts erm to obtain rm, signs the rm by computing a partial signature [30]

$$\sigma_{i_j} = rm^{2\Delta prk_{i_j}^{ev}}, \Delta = n_0!, \tag{1}$$

  where $prk_{i_j}^{ev}$ is the private key of ev_{i_j}.
- Each ev_{i_j} chooses a random number r, and computes a proof of correctness $\pi_{i_j} = (z, hv)$ [30] where

$$v' = v^r, rm' = rm^{4\Delta r}, hv = h(v, rm^{4\Delta}, v_{i_j}, \sigma_{i_j}^2, v', rm'),$$

$$z = prk_{i_j}^{ev} hv + r.$$

- Each ev_{i_j} returns $\mathsf{E}(puk_{dp}, \sigma_{i_j}||\pi_{i_j})$ to the DP.
- DP decrypts the encrypted message from each ev_{i_j} and verifies each partial signatures σ_{i_j} [30] by checking

$$hv \stackrel{?}{=} h(v, rm^{4\Delta}, v_{i_j}, \sigma_{i_j}^2, v^z v_{i_j}^{-hv}, rm^{4\Delta z} \sigma_{i_j}^{-2hv}).$$

- If all the verification passes, DP combines the t partial signatures to get a full signature [30] by computing

$$\sigma_{dp} = \prod_{j=1}^{t} \sigma_{i_j}^{2\gamma_{0,i_j}^{\mathcal{AU}}} \tag{2}$$

  where $\gamma_{0,i_j}^{\mathcal{AU}} = \Delta \prod_{j' \in \mathcal{AU} \setminus \{i_j\}} (0 - i_j) / \prod_{j' \in \mathcal{AU} \setminus \{i_j\}} (i_j - j')$. DP acquires the $cre = (rm, \sigma_{dp})$.

An IN registers to t AUs to acquire a cre similarly. In case of multiple data request, the IN sends multiple registration messages to AUs for multiple DACs. The IN with a set of investigation attributes $\mathcal{S}_{in}$ registers to their department au to obtain a secret key $sk_{in} \leftarrow \mathsf{KeyGen}(msk, \mathcal{S}_{in})$ and $\mathcal{SK}$. An IN initiates an investigation from his/her department, which will grant or deny the request and send a grant transaction $\text{Tx}^{\text{IniG}} =$ or a deny transaction Tx^{IniD} with a timestamp ts:

$$\text{Tx}_1^{\text{IniG}} = ("\text{IniGrant}", ReqID, CaseID, ts, \sigma_{au}),$$

$$\text{Tx}_1^{\text{IniD}} = ("\text{IniDeny}", ReqID, Null, ts, \sigma_{au}).$$

D. Data Uploading

The DP encrypts their data d to obtain $ed = \mathsf{E}(pub_{ev_i}, d)$ and sends a data message $dm = (cre, ed)$ to the au_i.

Upon receiving dm, au verifies its validity by computing

$$\sigma^e \stackrel{?}{=} rm^{4\Delta^2}.$$

If it passes, au sends a data uploading transaction to the CB:

$$\text{Tx}_2^{\text{Upl}} = ("\text{Upload}", cre, h(ed), ts, \sigma_{au_i}).$$

Next, the enclave ev_i produces a secure index as follows:

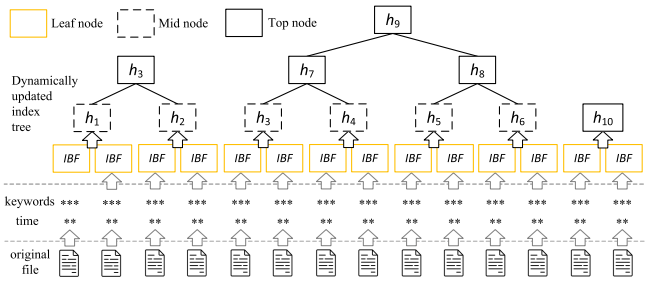


Fig. 4. Structure of dynamically updated index tree.

- Decrypt ed to obtain $d = \mathsf{D}(prk_{ev_i}, ed)$,
- Extract key information from d to acquire a set of items $\mathcal{IT} = \{it_i\}$ where each it_i has a set of keywords and time. For example, a keyword can be "assault", "murder", "theft", etc. Time can be "20221014".
- Initialize an IBF $\mathcal{B}_i$ and insert each keyword it_i into $\mathcal{B}$ by converting it into a prefix family $\mathcal{PF}_i = \{pr_{i_j}\}$ [32].
- Update $\mathcal{PF}_i$ by encoding each prefix into a keyword. For example, assume the keyword is "assault" and $\mathcal{PF}(4) = \{0100, 010*, 01**, ****\}$, encode each prefix in $\mathcal{PF}(4)$ into "assault:0100", "assault:010*", "assault:01**", "assault:0***", and "assault:****".
- Set for all j where rn_i being a random number:

$$\mathcal{B}_i[h_j(pr_{i_j})][h(h_{v+1}(h_j(pr_{i_j})) \oplus rn_i)] = 1,$$

$$\mathcal{B}_i[h_j(pr_{i_j})][1 - h(h_{v+1}(h_j(pr_{i_j})) \oplus rn_i)] = 0.$$

At the end of each consensus period, a winning enclave ev^* is selected. Each ev_i that has produced a secure index sends $(\mathcal{B}_i, rn_i)$ to ev^*. There are two cases for the index tree.

(1) If there is no index tree, ev^* produces an index tree (binary tree) $\mathcal{T}$ as follows.

- Sort the n_1 IBFs in a random order and prepare to build $\mathcal{T}$ from the bottom to up. Each IBF is called a leaf node.
- Assume that $\mathcal{B}_{new}$ is the father node of a left child node $\mathcal{B}_{le}$ and a right child node $\mathcal{B}_{ri}$, for $i \in [1, |\mathcal{B}|]$, set

$$\mathcal{B}_{new}[i][h(h_{v+1}(i) \oplus rn_{new})] =$$

$$\mathcal{B}_{le}[i][h(h_{v+1}(i) \oplus rn_{le})] \vee \mathcal{B}_{ri}[i][h(h_{v+1}(i) \oplus rn_{ri})].$$

- When the number of lead nodes reaches a predefined one, stop adding new leaf nodes and start producing a new tree (see Fig. 4 for an example).

(2) If there is already an index tree $\mathcal{T}$, ev_i appends the new IBFs to $\mathcal{T}$. After the tree construction, ev^* shares the new index tree(s) with other AUs and update the root hashes $\{rh_i\}$ in the ISC and send a root updating transaction:

$$\text{Tx}_3^{\text{Upd}} = ("\text{Update}", \{rh_i\}, \sigma_{au_i}).$$

E. Data Requesting

The IN has a request q consisting of a sequence of keywords and time periods tp and generates a data request dr as follows.

- Convert tp into a minimum set of prefixes $\mathcal{MS} = \{pr_i\}$ [32].

- Update $\mathcal{MS}$ by encoding each prefix into a keyword.
- For each $pr_i \in \mathcal{MS}$, compute

$$tk = \{(h_j(pr_i), h_{v+1}(h_j(pr_i)))\}_{j=1}^{v}. \quad (3)$$

- Compute key proof $kp = (puk_{in}, \mathsf{E}(puk_{ev_i}, prk_{in})$ and send $dr = (cre, tk, kp)$ to an AU, say au_i.

au_i first verifies the validity of dr. If it is not valid, au_i sends a request deny transaction to the CB; otherwise, it sends a request grant transaction:

$$\mathrm{Tx}_4^{\mathrm{ReqD}} = (''\mathrm{ReqDeny}'', cre, h(tk), ts, \sigma_{au_i}),$$
$$\mathrm{Tx}_4^{\mathrm{ReqG}} = (''\mathrm{ReqGrant}'', cre, h(tk), ts, \sigma_{au_i}).$$

Given tk, au_i locally searches from the root $\mathcal{B}_{rt}$ of $\mathcal{T}$ and checks whether there exists a pr_i that satisfies for $1 \leq j \leq v$:

$$\mathcal{B}_{rt}[H_j(pr_i)][h(h_{v+1}(h_j(pr_i)) \oplus rn)] = 1.$$

If so, au_i processes tk against the left child node and right child node of the root. The process recursively applies to subtrees and stops when au_i determines tk matches no leaf nodes or at least one leaf node. au_i generates Merkle proofs $\mathcal{MP}$ for the matching node.

Given a matching leaf node $\mathcal{B}$, we assume that au_i has stored the corresponding data ed locally. We will discuss in Section VIII about how we respond to IN if au_i does not have the requested data. Next, ev_i produces watermarked-and-encrypted data er as follows.

- Decrypt ed and kp to obtain data d and prk_{in}.
- Verify prk_{in} by checking if $g^{prk_{in}} = puk_{in}$.
- Code prk_{in} by using an improved embedding method to guarantee the recovery rate in traitor tracing. (Refer to VII.A for details on improvements.)
- Embed prk_{in} into d by computing $d' = \mathsf{Embed}(key, prk_{in}||d)$.
- Determine the access structure $\mathbb{A}$ for d and encrypt d' by computing $er = \mathsf{Encrypt}(mpk, \mathbb{A}, d')$.

au_i sends $(er, \mathcal{MP})$ back to the IN. The IN verifies er and then decrypts er by computing $rr = \mathsf{Decrypt}(pk, er, sk_{in})$.

F. Analyzing, Reporting, and Trailing

Since this part is not our focus, we only provide a process sketch of it. After collecting enough data for the investigation, the IN carries out a professional and systematic investigation in cooperation with other forensics professionals. At the end of the investigation, the IN finalizes a When an investigation report rp and sends a reporting transaction to the CB: $\mathrm{Tx}_5^{\mathrm{Rep}} = [''\mathrm{Report}'', h(rp), ts, \sigma_{in})$.

Afterwards, the court holds a trial handle down a verdict ve and close the case. Finally, the judge sends a judging transaction to the CB: $\mathrm{Tx}_6^{\mathrm{Jud}} = [''\mathrm{Judge}'', h(ve), ts, \sigma_{co}]$.

G. Traitor Tracing

In this work, we assume that INs initiates a partial-data leakage attack by cropping a requested picture and leaking the cropped data. When the leaked data pwd is found, any enclave will extract a watermark wm from pwd by computing $wm = \mathsf{Extract}(key, pwd)$. Then, the enclave computes its corresponding public key to track the traitor.

The successful tracking probability depends on how much the traitor crops the picture as well as the length of the secret key. We will analyze it in Section VII.

VI. Privacy and Security Analysis

A. Privacy

1) Anonymity: Theorem 1. Themis provides anonymity under the indistinguishability of the asymmetric encryption scheme Ω.

Proof. For DPs, they register to t AUs to obtain t encrypted partial signatures $\sigma_1, \sigma_2, \ldots, \sigma_t$ that are further combined to be a full signature σ. Intuitively, the anonymity of DPs is guaranteed by protecting the link between $\sigma_1, \sigma_2, \ldots, \sigma_t$ and σ. Here, each set of $\{\sigma_{i1}, \sigma_{i2}, \ldots, \sigma_{it}\}$ is secured by E and sent from the t AUs to a DP.

AU or a colluding IN as the $\mathcal{A}$. The view of AU and the view of IN towards the message of a DP is the same because they can only observe DP's communication channel. Although AU processes DP's data messages, it cannot access the contents due to the secure enclave. We define two events for proving anonymity. (1) Event 11: $\mathcal{A}$ successfully guesses k in $\mathsf{Exp}_{\mathcal{A},\Pi}^{\text{Any-DP}}(\lambda_0)$ (Section IV-C) by successfully breaking the asymmetric encryption Ω. (2) Event 21: $\mathcal{A}$ successfully guesses k in $\mathsf{Exp}_{\mathcal{A},\Pi}^{\text{Any-DP}}(\lambda_0)$ by randomly guessing the correct k [41]. We have the following inequation:

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A},\Pi}^{\text{Any-DP}}(\lambda_0) &= \left|\Pr[\text{Event 11}] + \Pr[\text{Event 12}] - \frac{1}{n_1}\right| \\ &= \left|\Pr[\mathrm{PubK}_{\mathcal{A},\Omega}(\lambda_0) = 1]^{t+1} + \frac{1}{n_1} - \frac{1}{n_1}\right| \\ &= \Pr[\mathrm{PubK}_{\mathcal{A},\Omega}(\lambda_0) = 1]^{t+1} \leq \mathrm{negl}(\lambda_0).\end{aligned}$$

For INs, the proof is similar such that $\mathsf{Adv}_{\mathcal{A},\Pi}^{\text{Any-IN}}(\lambda_0) = \frac{1}{2^{|\lambda_0|}} \leq \mathrm{negl}(\lambda_0)$. We have proved the anonymity goal in Definition 2.

2) Unlinkability: Theorem 2. Themis provides unlinkability under the indistinguishability of Ω and Discrete Logarithm (DL) assumption.

Proof. For DPs, $\mathcal{A}$ can break their unlinkability by either linking d_0 and d_1 in $\mathsf{Exp}_{\mathcal{A},\Pi}^{\text{Unl-DP}}(\lambda_0, \lambda_1)$ or linking cre_{dp} and cre'_{dp}. Meanwhile, we require that DPs register to different AUs for a new DAC and assume that the number of AUs in the new registration is t^* $(0 \leq t^* < t)$. We define two experiments for proving unlinkability. (1) Event21: $\mathcal{A}$ successfully guesses k in $\mathsf{Exp}_{\mathcal{A},\Pi}^{\text{Unl-DP}}(\lambda_0, \lambda_1)$ (Section IV-C) by successfully breaking the asymmetric encryption Ω at DP_k. (2) Event 22: $\mathcal{A}$ successfully guesses k in $\mathsf{Exp}_{\mathcal{A},\Pi}^{\text{Unl-DP}}(\lambda_0, \lambda_1)$ by breaking the signature scheme Γ t^* times. (3) $\mathcal{A}$ successfully guesses k in $\mathsf{Exp}_{\mathcal{A},\Pi}^{\text{Unl-DP}}(\lambda_0, \lambda_1)$ by randomly guessing the correct k. In Event 22, $\mathcal{A}$ has to locate the t^* new partial signatures first and then break their signing

keys. Therefore, we have the following equation:

$$\begin{aligned}&\mathsf{Adv}^{\text{Unl-DP}}_{\mathcal{A},\Pi}(\lambda_0,\lambda_1)\\&=\left|\Pr[\text{Event 21}]+\Pr[\text{Event 22}]+\Pr[\text{Event 23}]-\frac{1}{n_1}\right|\\&=\left|\Pr[\text{Event 21}]+\Pr[\text{Event 22}]+\frac{1}{n_1}-\frac{1}{n_1}\right|\\&=\frac{\Pr[\text{PubK}_{\mathcal{A},\Omega}(\lambda_0)=1]}{n_1}+\frac{\Pr[\text{Sig-forge}_{\mathcal{A},\Gamma}(\lambda_1)=1]}{(\mathrm{C}_t^{t*})^2}\\&\leq \mathsf{negl}(\lambda_0)+\mathsf{negl}(\lambda_1),\end{aligned}$$

which is negligible. For INs, the proof is similar such that $\mathsf{Adv}^{\text{Unl-IN}}_{\mathcal{A},\Pi}(\lambda_0,\lambda_1)\leq \mathsf{negl}(\lambda_0)+\mathsf{negl}(\lambda_1)$. We have proved the anonymity goal in Definition 3.

B. Security

1) Private Access Control: Theorem 3. Themis provides private access control under the indistinguishability of Ω and the DLIN assumption on asymmetric pairing groups in the random oracle model [20].

Proof. Based on $\mathsf{Exp}^{\text{PAC-AU}}_{\mathcal{A},\Pi}(\lambda_0,\lambda_2)$, from the AU's view, the uploaded data is encrypted by the DP and stored locally. If the data is requested by an IN, its ciphertext is processed within enclave, watermarked, and encrypted by the public key pk. We define two events for proving private access control. (1) Event 31: $\mathcal{A}$ successfully guesses b in $\mathsf{Exp}^{\text{PAC-AU}}_{\mathcal{A},\Pi}(\lambda_0,\lambda_2)$ (Section IV-C) by successfully breaking the asymmetric encryption Ω. (2) Event 32: $\mathcal{A}$ successfully guesses b in $\mathsf{Exp}^{\text{PAC-AU}}_{\mathcal{A},\Pi}(\lambda_0,\lambda_2)$ by successfully breaking the ABE Υ. (3) Event 33: $\mathcal{A}$ successfully guesses b in $\mathsf{Exp}^{\text{PAC-AU}}_{\mathcal{A},\Pi}(\lambda_0,\lambda_2)$ by randomly guessing the correct b. Therefore, we have the following equation:

$$\begin{aligned}&\mathsf{Adv}^{\text{PAC-AU}}_{\mathcal{A},\Pi}(\lambda_0,\lambda_2)\\&=\left|\Pr[\text{Event 31}]+\Pr[\text{Event 32}]+\Pr[\text{Event 33}]-\frac{1}{2}\right|\\&=\Pr[\text{PubK}_{\mathcal{A},\Omega}(\lambda_0)=1]+\Pr[\text{PubK}_{\mathcal{A},\Upsilon}(\lambda_0)=1]\\&\leq \mathsf{negl}(\lambda_0)+\mathsf{negl}(\lambda_2).\end{aligned}$$

Based on $\mathsf{Exp}^{\text{PAC-IN}}_{\mathcal{A},\Pi}(\lambda_0,\lambda_2)$, the IN's view is a little different for possessing a secret key $sk_{\mathcal{A}}$ associated with $\mathcal{S}_{\mathcal{A}}$. We have the following equation:

$$\begin{aligned}\mathsf{Adv}^{\text{PAC-IN}}_{\mathcal{A},\Pi}(\lambda_0,\lambda_2)&\leq\frac{1}{2^{|\lambda_0|}}+(8Q+2)\mathsf{Adv}^{\mathcal{B}}_{\text{DLIN}}(\lambda_2)+\frac{(16Q+6)}{p}\\&=\mathsf{negl}(\lambda_0)+\mathsf{negl}(\lambda_2),\end{aligned}$$

where Q is the number of queries and $p=\Theta(\lambda_2)$ is the order of the pairing group. Please refer to [20] for details. We have proved the private access control goal.

2) Authentication: Theorem 4. Themis provides authentication under the security of the standard RSA signature scheme for $t=w+1$ in the random oracle model for h, where w is the number of corrupted AUs [30].

TABLE III
KEY EXPERIMENTAL PARAMETERS

Parameter	Value	Parameter	Value
n_0	4	n_1	$[100, 1000]$
n_2	$[100, 500]$	$\lvert prk^1\rvert$, $\lvert puk^1\rvert$	512, 512
$\lvert prk^2\rvert$, $\lvert puk^2\rvert$	1024	$\lvert mpk\rvert$, $\lvert msk\rvert$	512, 512
$\lvert sk\rvert$	512	$\lvert key\rvert$	512
$\lvert k_i\rvert$	256	v	5
h	SHA256%2	L	10000

Proof. We have asked the AUs to register DPs and INs by signing their submitted message and generating a partial signature. Assume that there are w corrupted AUs and they try to forge a valid full signature σ given their w secret shares. We define one event for proving authentication. Event 4: $\mathcal{A}$ successfully forges a full signature in $\mathsf{Exp}^{\text{Aut}}_{\mathcal{A},\Pi}(\lambda_1)$ (Section IV-C) by successfully breaking the signature scheme Γ for any uncorrupted AU from n_0-w uncorrupted AUs. Therefore, we have the following equation:

$$\begin{aligned}&\mathsf{Adv}^{\text{Aut}}_{\mathcal{A},\Pi}(\lambda_1)=\Pr[\text{Event 4}]\\&\leq(n_0-w)\Pr[\text{Sig-forge}_{\mathcal{A},\Gamma}(\lambda_1)=1]\leq\mathsf{negl}(\lambda_1).\end{aligned}$$

We have proved the authentication goal in Definition 4.□

We also compare Themis with existing work on model, privacy, security, and function in Table II. Only the proposed Themis provides all the design objectives.

VII. PERFORMANCE EVALUATION

A. Experiment Settings

Dataset. Since there are few sources of datasets on data uploading or requesting in digital forensics, we synthesized two plausible groups (DPs and INs) to simulate the uploading and requesting activities. Specifically, we created n_1 DPs and n_2 INs. Both of the two groups uploaded/requested data spontaneously in a predefined time period. n_1 and n_2 were drawn from two ranges to test scalability.

Parameters. We list the key experimental paraments in Table III, including the number of AUs n_0, number of DPs n_1, number of INs n_2, private key and public key (prk^1, puk^1) for signature scheme, private key and public key (prk^2, puk^2) for DPs and INs, public key and master secret key (mpk, msk), secret key sk, key, k_i $(1\leq i\leq v+1)$, and hash function h_i $(1\leq i\leq v+1)$, h.

Metrics. We evaluated the computational costs and communication overhead for DP, IN, AU, and EV. Since Themis is a decentralized framework, we have chosen Ethereum to be the underlying platform and evaluate the basic blockchain performance including consensus time, transaction confirmation time, and gas costs. We also evaluated the scalability (response time for INs) and traceability (private key recovery rate after partial-data leakage) of Themis.

Setup. The implementation details of Themis are shown in Fig. 5. The hardware settings are listed in Table IV. We initialized

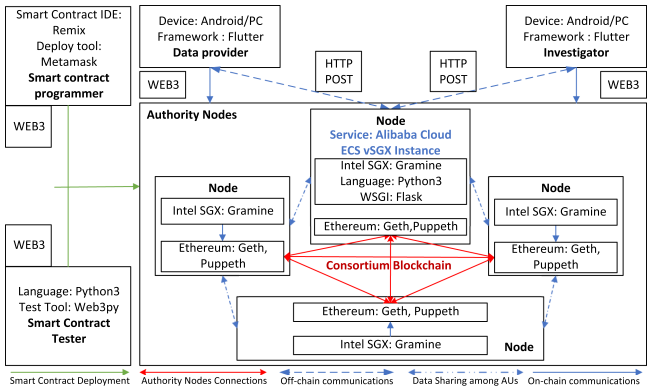


Fig. 5. Implementation details of DDF.

TABLE IV
HARDWARE SETTINGS

Smartphone (Huawei Mate 40) as DP and IN	
CPU	Kirin 9000E
Memory	8 GB
Operating System	HarmonyOS
Server (ECS) as AU	
CPU	Intel Xeon 8269CY @2.50GHz
Memory	4 GB
Operating System	Ubuntu 20.04.6 LTS

Themis by installing SGX Driver, SGX SDK, SGX PSW, SGX DCAP, and Gramine on the AU's server, deploying the web application into enclave through Gramine. The blockchain test platform was Ethereum 2.0, and Geth was used to establish the Ethereum network. We instantiated four AU nodes, each with an Ethereum account, on four Alibaba Cloud ECS servers supporting SGX 2.0. The nodes communicated with each other through the LAN. We instantiated DP/IN on two smartphones, created a cross-platform application based on the Flutter framework. They communicated with AU through the HTTP protocol and interact with CB through the web3 protocol. We used Puppeth to create the genesis block. The consensus mechanism was Clique with a block creation time of 10 seconds. After the consortium blockchain started running, each blockchain node generated the enode information. We wrote the smart contract using Remix and deployed the smart contract through MetaMask. We recorded the generated contract address and application binary interface, and then invoked the contract through the web3 protocol. We have uploaded the codes to https://github.com/UbiPLab/Themis.

For *watermark embedding*, we transform the 1024-bit public key into a symmetrical watermark by using even-parity Hamming encoding, padding, spread-spectrum encoding, XORing, double sampling, and flipping. We convert the host image to the YCrCb space and extract the Y component as the luminance matrix. We set the global embedding intensity to two and change each element of the luminance matrix to the initial value plus the global embedding intensity multiplied by the value of each element in the symmetrical watermark. We convert the new luminance matrix and the original host image's Cr and Cb components to an RGB image to generate the watermarked image. For *watermark extracting*, we convert the watermarked image to the YCrCb space and extract the Y component as the luminance matrix. We use the local variance of the luminance matrix to calculate the mean square error minimization estimate of the watermark module to obtain the watermark module and then synchronize the watermark based on symmetry. We determine the watermark state based on the central limit theorem, decode the watermark and the mask matrix to obtain the public key information after group Hamming encoding, and then recover the original public key through Hamming decoding.

B. Computational Costs

We analyze the computational costs for DP, IN, AU, and EV through counting their number of operations. We denote G_x as a general generation function to obtain x.

A DP's computational costs come from entity registration and data uploading, which includes: compute a public key $pubk_{dp}$, encrypt a registration message rm, decrypt t encrypted messages $\mathsf{E}(puk_{dp}, \sigma_{i_j}||\pi_{i_j})$, verify and combine t partial signatures $\{\sigma_i\}$, and encrypt data d. i.e., $(ex + \mathsf{E} + \mathsf{D} + (6ex + 2mu) + t * ex + (t-1)mu) + E$. The two phases cost the DP 0.69 s and 1.17 s.

An IN's computational cost mainly stem from entity registration and data requesting, which includes: the same operations as DP in entity registration, generate a token tk, compute a key proof kp, and verify and decrypt er. i.e., $(ex + \mathsf{E} + \mathsf{D} + (6ex + 2mu) + t * ex + (t-1)mu) + \mathsf{TokenGen} + \mathsf{E} + ex + \mathsf{D}$. The two phases cost the IN 0.77 s and 16.32 s.

An AU's computational cost originate from system initiation, data uploading, and data requesting, which includes: initialize the CB, verifies n_1 dms, verifies n_2 drs, n_2 searches on the index tree, and generate Merkle hash proofs $\{pf_i\}$. i.e., $\mathsf{Ini}_{\mathsf{CB}} + 2n_1ex + 2n_2ex + \mathsf{Search}(\mathcal{T}, tk) + \mathsf{G}_{\{pf_i\}}$. The three phases cost the AU 64 ms, 4.82 s, and 5.12 s.

An EV's computational costs are from system initiation, entity registration, data uploading, data requesting, and traitor tracing, which includes: generate all the keys and functions, decrypt $n_1 + n_2$ encrypted registration messages, sign $n_1 + n_2$ registration messages, generate $n_1 + n_2$ proofs, decrypt n_1 encrypted data, generate n_1 indexes, decrypt n_2 encrypted data and key proofs, verify n_2 secret keys, embed n_2 times, encrypt n_2 times, and extract the key. i.e., $(\mathsf{Gen} + \mathsf{Setup} + \mathsf{KGen} + \mathsf{Initialize}) + (n_1 + n_2)(\mathsf{D} + 5ex + h) + (n_1(\mathsf{D} + \mathsf{IndexGen}) + \mathsf{G}_{\mathcal{T}}) + n_2(2\mathsf{D} + ex + \mathsf{Embed} + \mathsf{Encrypt}) + \mathsf{Extract}$.

The five phases cost the IN 0.93 s, $25 * (n_1 + n_2)$ ms, $6.88n_1 + 0.92$ s, $7.59n_2$ s, and 0.17 s. The main measured values for execution time of each entity are listed in Table V.

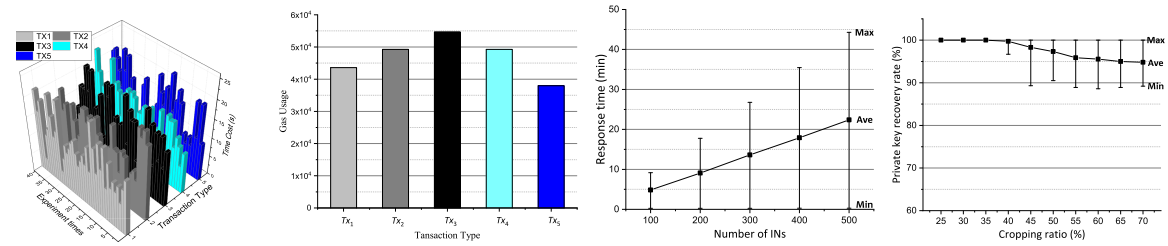
C. Communication Overhead

We analyze the communication overhead for the four main entities. A DP submits t encrypted registration messages and a data message, i.e., $t|erm| + |cre| + |\mathsf{E}| = 4.45$ MB. An IN submits t encrypted registration messages, a data message, and a data request dr. i.e., $t|erm| + |dr| = 0.25$ KB. Assume that all requests are valid and matched

TABLE V
COMPARISON OF COMPUTATIONAL COSTS

Scheme	System Initialization		Entity Registration			Data Uploading			Data Requesting			Traitor Tracing
	AU	EV	DP	IN	EV	DP	AU	EV	IN	AU	EV	EV
[16]	$\mathsf{Ini}_{\mathsf{B}}^{1a}$	n/a	G_{cer}^{1b}	G_{cer}^{1b}	n/a	$h+\mathsf{G}_\sigma$	$n_1(h+\mathsf{G}_\sigma)$	n/a	n/m^{1c}	n/a	n/a	n/a
[17]^{2a}	$\mathsf{Ini}_{\mathsf{B}}+\mathsf{Gen}+\mathsf{Setup}$	n/a	ex	ex	n/a	$\mathsf{Encrypt}+\mathsf{G}_{root}+\mathsf{G}_\sigma^{2b}$	$n_1\mathsf{Ver}_\sigma$	n/a	$H+4\mathsf{G}_\sigma+\mathcal{N}*\mathsf{G}_{bres}+\mathsf{PIR}+\mathsf{G}_{SK}+\mathsf{Decrypt}^{2c}$	$n_2(\mathsf{Ver}_\sigma+\mathsf{G}_{res}+\mathsf{G}_\pi+\mathsf{G}_\sigma+4\mathsf{Ver}_\sigma)^{2d}$	n/a	n/a
[15]^{3a}	$\mathsf{Ini}_{\mathsf{CB}}+\mathsf{Gen}_{cre}+\mathsf{Setup}$	n/a	$2ex+mu+\mathsf{G}_\pi+\mathsf{G}_\sigma$	$2ex+mu$	$n_1\mathsf{Ver}_{Tx_1}+n_2\mathsf{KGen}+n_2\mathsf{G}_\sigma+n_2\mathsf{Ver}_{Tx_2}^{3b}$	$\mathsf{Encrypt}+\mathsf{G}_{cre}+\pi'+H_1+H_2+\mathsf{G}_\sigma$	$n_1\mathsf{Ver}_{Tx_3}$	n/a	$\mathsf{G}_{\pi'}+\mathsf{G}_{\pi''}+\mathsf{G}_{cre}+2\mathsf{G}_\sigma+\mathsf{OT}+\mathsf{Dec}+\mathsf{Decrypt}$	$n_2\mathsf{Ver}_{Tx_4}+\mathsf{G}_\sigma+\mathsf{Encrypt}$	n/a	n/a
Themis	$\mathsf{Ini}_{\mathsf{CB}}$	$\mathsf{Gen}+\mathsf{Setup}+\mathsf{KGen}+\mathsf{Initialize}$	$(t+7)ex+(t+1)mu+\mathsf{E}+\mathsf{D}$	$(t+7)ex+(t+1)mu+\mathsf{E}+\mathsf{D}$	$(n_1+n_2)(\mathsf{D}+5ex+h)$	E	$2n_1ex$	$n_1(\mathsf{D}+\mathsf{IndexGen})+\mathsf{G}_\mathcal{T}$	$\mathsf{TokenGen}+\mathsf{E}+ex+\mathsf{D}$	$n_2(2ex+\mathsf{Search}+\mathsf{G}_{\{pf_i\}})$	$n_2(2\mathsf{D}+ex+\mathsf{Embed}+\mathsf{Encrypt})$	Extract

1a: No indication of what blockchain. 1b: register to certification authority using vehicular public key infrastructure (IEEE 1609.2). 1c: Not mentioned.
2a: The function and notion that are same to ours do not always mean the same. 2b: root is a Merkle root computed from consecutive ciphertexts.
2c: DP leverages the private information retrieval (PIR) scheme [43] to retrieve the data from a distributed data storage system.
2d: AU computes a zero knowledge proof π by using the technique in [44] to approve that SK is indeed computed based on the authorized res.
3a: Same to 2a; 3b: AU conducts these operations.



(a) Tx Confirmation Time (b) Gas Costs (c) Response Time (d) Traceability

Fig. 6. Performance of themis.

with stored data, an AU sends n_2 grant transactions, n_1 data uploading transactions, n_2 request grant transactions, and n_2 results, i.e., $n_1|\mathrm{Tx}^{\mathrm{Upl}}|+n_2(|\mathrm{Tx}^{\mathrm{IniG}}|+|\mathrm{Tx}^{\mathrm{ReqG}}|+|er|+|\mathsf{G}_{\{pf_i\}}|)=0.43n_1$ KB + $4.41n_2$ MB. An EV sends public parameters, n_1+n_2 partial signatures and proofs, n_1 indexes and random numbers, n_2 encrypted results, and a public key of traitor IN. i.e., $|pp|+(n_1+n_2)(|\sigma|+|\pi|)+n_1(|\mathcal{B}|+|rn|)+n_2|er|+|pubk_{in}|=0.22n_1$ KB + $4.41n_2$ MB.

D. Basic Blockchain Performance

We set the block period to be 20 seconds. The real consensus time fluctuates around 20 seconds because of the hardware interference. The interval time between two transactions of six types is 1 s. Due to network delay and consensus mechanism, the average transaction confirmation time of transactions is approximately 13.34 seconds as shown in Fig. 6(a). We measure the gas cost of six types of transactions in Fig. 6(b). The average gas cost for Tx_1 is about 43572. We set the gas price to be 1 Gwei (0.000000001 Ether) in the Themis blockchain, at the time of writing (May 1st, 2023), the exchange rate is \$1,870.76 USD per Ether. Each Tx_1 costs about 0.000043572 Ether (0.08 USD).

E. Scalability

Scalability here refers to the response time of an IN between data requesting and result receiving, including blockchain search time and communication delay. We evaluate the maximum, minimum, and average response time. In Fig. 6(c), the average response time increases with the number of INs, which is less than 5 minutes when $n_2=100$.

F. Traceability

The private key recovery rate (PKRR) of the watermarking module is affected by the cropping ratio cr. We set the ratio from 25% to 70% and randomly crop out a square in three pictures twenty times for each ratio. After watermark extraction, we compute how many bits of the IN's private key prk_{in} are recovered and record the average private key recovery rate. Results in Fig. 6(d) show that we can recover prk_{in} without error when $cr\leq 35\%$. When cr is 70%, the average PKRR is 94.8%.

G. Comparison With Existing Work

Since the existing work do not have EV in their system model, we incorporate the computational cost and communication overhead into the one of AU. The comparison results are listed in Tables V and VI. Although Themis does not exceed other schemes, still it empowers DPs and INs to securely share data against multiple attacks.

TABLE VI
COMPARISON OF COMMUNICATION OVERHEAD

Scheme	System Initialization		Entity Registration			Data Uploading			Data Requesting			Traitor Tracing
	AU	EV	DP	IN	EV	DP	AU	EV	IN	AU	EV	EV
[16]	$\lvert\mathsf{Ini}_{\mathsf{B}}^{1a}\rvert$	n/a	$\lvert id\rvert$	$\lvert id\rvert$	n/a	$\lvert m\rvert^{1b}+$ $\lvert h\rvert+\lvert\sigma\rvert$	$\lvert m\rvert+$ $\lvert h\rvert+\lvert\sigma\rvert$	n/a	n/m	n/a	n/a	n/a
[17]	$\lvert PK\rvert$	n/a	$\lvert pk\rvert$	$\lvert pk\rvert$	n/a	$\lvert Tx^1\rvert$	n/a	n/a	$\lvert req\rvert+\lvert Tx^2\rvert+$ $\mathcal{N}*\lvert bres\rvert+$ $\lvert\mathsf{PIR}\rvert+\lvert Tx^4\rvert+$ $\lvert Tx^5\rvert+\lvert Tx^6\rvert$	$n_2(\lvert res\rvert+$ $Tx^3)$	n/a	n/a
[15]	$\lvert pp.\rvert$	n/a	$\lvert Tx^1\rvert$	$\lvert\mathbb{A}\rvert^{3a}$	$n_2(\lvert sk\rvert+$ $\lvert Tx_2\rvert)^{3b}$	$\lvert Ed\rvert+$ $\lvert Tx_3\rvert^{3c}$	n/a	n/a	$\lvert Tx_4\rvert+\lvert id\rvert+$ $\lvert pk\rvert+\lvert L\rvert+$ $2\lvert\sigma\rvert+\lvert\mathsf{OT}\rvert^{3d}$	$n_2(\lvert Tx_5\rvert+$ $\lvert\mathsf{Enc}\rvert)$	n/a	n/a
Themis	$n_2\lvert\mathsf{Tx}^{\mathsf{IniG}}\rvert$	$\lvert pp\rvert$	$t\lvert erm\rvert$	$t\lvert erm\rvert$	(n_1+n_2) $(\lvert\sigma\rvert+\lvert\pi\rvert)$	$\lvert cre\rvert+$ $\lvert\mathsf{E}\rvert$	$n_1\lvert\mathsf{Tx}^{\mathsf{Upl}}\rvert$	$n_1(\lvert\mathcal{B}\rvert+$ $\lvert rn\rvert)$	$\lvert dr\rvert$	$n_2(\lvert\mathsf{Tx}^{\mathsf{ReqG}}$ $+\lvert er\rvert+\lvert\mathcal{MP}\rvert)$	$n_2\lvert er\rvert$	$\lvert pubk_{in}\rvert$

$1a$: Blockchain creation related messages; no indication of what messages to send. $1b$: messages to send; provide only integrity and authentication.
$3a$: Set of attributes. $3b$: AU sends the grant transaction. $3c$: Ed is encrypted data. $3d$: L is a download link for data retrieval.

VIII. DISCUSSIONS

A. Data Sharing Among AUs

If au_i does not have what an IN requests, we ask the AUs to share encrypted data in a peer-to-peer distributed file system, such as InterPlanetary File System (IPFS) [44]. au_i sends back the identity of the AU that holds the corresponding data and an permission token tk to the IN. The IN passes tk to the AU, say au_j, which will return the data.

B. Malicious Data Providers

Since DDF gathers data in a crowdsourcing way, it is likely that some DPs are malicious and uploads falsified data. Such data are fatal to investigations as they pretend to be authentic data that cloud INs' judgement. We have two countermeasures in this case. First, we can leverage some detection mechanisms running on the DDS apps regarding image [45] and video [46]. Second, once a data forgery is detected, we can adopt conditional privacy-preserving authentication [19], [47] to reveal the identity of the data provider to revoke their login permission. They will stand as the defense forefront of data uploading.

C. Watermarking

The watermarking technique that we improved in this work is applicable to more than just decentralized digital forensics. For example, it can be used for copyright tracking, transfer recording, and access control. First, a digital watermark embedded in a pictorial work speaks for its owner and copyright. Second, when a watermarked item is transferred among different entities and embedded with a new watermark upon each transfer, the watermarks can be considered as a proof of transfer record. Third, we can embed several watermarks of different data users into an item to claim that such an item are only accessible to these data users. For the last two scenarios, we have to preset different areas for different watermarks to avoid extraction failures.

D. Applicability

There are many real-world use cases where the proposed DDF approach is useful to transform and enhance existing digital forensics techniques, such as Criminal Digital Forensics (CDF) and Vehicular Digital Forensics (VDF). In CDF as well as VDF, the collection of data is key to cracking a criminal case and determining liability. Our approach transforms existing DF techniques into collecting real-time data from multiple data sources in a distributed manner. In CDF, the management of potential evidence uploaded by data providers, e.g., determining which crime investigator can access the uploaded data and which crime investigator leaks data, is extremely important to advance the investigation process and fight crimes. Our approach enhances existing DF techniques by guaranteeing that only qualified data users can access corresponding data and prevent internal adversaries from leaking valuable evidence.

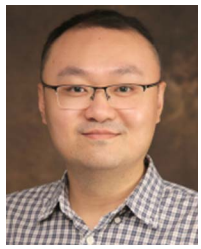
IX. CONCLUSION

In this paper, we have proposed decentralized digital forensics based on TEE and blockchain to achieve traceable anonymity, private access control, and efficient search. To achieve the three goals, we integrate secure and robust watermarking scheme with decentralized anonymous authentication, realize fine-grained access control against malicious authorities, and facilitate efficient and secure search with a dynamically updated data structure. We formally define and prove the privacy and security. Experimental results exhibit its good practicability and efficiency.

REFERENCES

[1] World Health Organization, injuries and violence, 2021. [Online]. Available: http://www.who.int/news-room/fact-sheets/detail/injuries-and-violence
[2] G. G. Richard III and V. Roussev, "Next-generation digital forensics," *Commun. ACM*, vol. 49, no. 2, pp. 76–81, 2006.
[3] S. L. Garfinkel, "Digital forensics research: The next 10 years," *Digit. Investigation*, vol. 7, pp. S64–S73, 2010.
[4] L. Caviglione, S. Wendzel, and W. Mazurczyk, "The future of digital forensics: Challenges and the road ahead," *IEEE Secur. Privacy*, vol. 15, no. 6, pp. 12–17, Nov./Dec. 2017.
[5] United Nations, Crime Prevention, 2023. [Online]. Available: https://www.unodc.org/unodc/en/justice-and-prison-reform/cpcj-crimeprevention-home.html
[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[7] H. -N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8076–8094, Oct. 2019.
[8] Cooperation with United Nations entities, 2023. [Online]. Available: https://www.interpol.int/Our-partners/International-organization-partners/INTERPOL-and-the-United-Nations/Cooperation-with-United-Nations-entities
[9] A. Kiayias and Q. Tang, "Traitor deterring schemes: Using bitcoin as collateral for digital content," in *Proc. 22nd ACM Conf. Comput. Commun. Secur.*, Denver, USA, 2015, pp. 231–242.

[10] J. Frankle, S. Park, D. Shaar, S. Goldwasser, and D. Weitzner, "Practical accountability of secret processes," in *Proc. 27th USENIX Secur. Symp.*, Baltimore, USA, 2018, pp. 657–674.
[11] M. Li, Y. Chen, C. Lal, M. Conti, F. Martinelli, and M. Alazab, "Nereus: Anonymous and secure ride-hailing service based on private smart contracts," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 4, pp. 2849–2866, Jul./Aug. 2023, doi: 10.1109/TDSC.2022.3192367.
[12] W. Yang, Y. Geng, L. Li, X. Xie, and L. Huang, " Achieving secure and dynamic range queries over encrypted cloud data," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 107–121, Jan. 2022.
[13] M. Li, Y. Chen, S. Zheng, D. Hu, C. Lal, and M. Conti, "Privacy-preserving navigation supporting similar queries in vehicular networks," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 2, pp. 1133–1148, Mar./Apr. 2022, doi: 10.1109/TDSC.2020.3017534.
[14] M. Li et al., "Astraea: Anonymous and secure auditing based on private smart contracts for donation systems," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 4, pp. 3002–3018, Jul./Aug. 2023, doi: 10.1109/TDSC.2022.3204287.
[15] M. Li, Y. Chen, C. Lal, M. Conti, M. Alazab, and D. Hu, "Eunomia: Anonymous and secure vehicular digital forensics based on blockchain," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 1, pp. 225–241, Jan./Feb. 2023, doi: 10.1109/TDSC.2021.3130583.
[16] M. Cebe, E. Erdin, K. Akkaya, H. Aksu, and S. Uluagac, "Block4Forensic: An integrated lightweight blockchain framework for forensics applications of connected vehicles," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 50–57, Oct. 2018.
[17] M. Li, J. Weng, J.-N. Liu, X. Lin, and C. Obimbo, "Towards vehicular dorensics from decentralized trust: An accountable, privacy-preservation, and secure realization," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 7009–7024, May 2022.
[18] C. Garman, M. Green, and I. Miers, "Decentralized anonymous credentials," in *Proc. 21st Netw. Distrib. Syst. Secur. Symp.*, San Diego, USA, 2014, pp. 1–15.
[19] D. Maram et al., "CanDID: Can-do decentralized identity with legacy compatibility, sybil-resistance, and accountability," in *Proc. IEEE 42nd Symp. Secur. Privacy*, 2021, pp. 1348–1366.
[20] S. Agrawal and M. Chase, "FAME: Fast attribute-based message encryption," in *Proc. 24th ACM Conf. Comput. Commun. Secur.*, Dallas, USA, 2017, pp. 665–682.
[21] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Comput.*, vol. 16, no. 1, pp. 69–73, Jan./Feb. 2012.
[22] Z. Shi, X. Fu, X. Li, and K. Zhu, "ESVSSE: Enabling efficient, secure, verifiable searchable symmetric encryption," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 7, pp. 3241–3254, Jul. 2022.
[23] H. Shafagh, L. Burkhalter, S. Ratnasamy, and A. Hithnawi, "Droplet: Decentralized authorization and access control for encrypted data streams," in *Proc. 29th USENIX Secur. Symp.*, Boston, USA, 2020, pp. 2469–2486.
[24] S. Dougherty, R. Tourani, G. Panwar, R. Vishwanathan, S. Misra, and S. Srikanteswara, "APECS: A distributed access control framework for pervasive edge computing services," in *Proc. 28th ACM Conf. Comput. Commun. Secur.*, Seoul, South Korea, 2021, pp. 1405–1420.
[25] C. Zhang, C. Xu, J. Xu, Y. Tang, and B. Choi, "GEM2-Tree: A gas-efficient structure for authenticated range queries in blockchain," in *Proc. IEEE 35th Int. Conf. Data Eng.*, Macao, China, 2019, pp. 842–853.
[26] C. Xu, C. Zhang, and J. Xu, "vChain: Enabling verifiable Boolean range queries over blockchain databases," in *Proc. Int. Conf. Manage. Data*, Amsterdam, Netherlands, 2019, pp. 141–158.
[27] F. McKeen et al., "Intel software guard extensions (Intel SGX) support for dynamic memory management inside an enclave," in *Proc. 5th Int. Workshop Hardware Architectural Support Secur. Privacy*, Seoul, South Korea, 2016, pp. 1–9.
[28] Intel, Which platforms support Intel software guard extensions (Intel SGX) SGX2?, 2023. [Online]. Available: https://www.intel.com/content/www/us/en/support/articles/000058764/software/intel-security-products.html
[29] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," 2014. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf
[30] V. Shoup, "Practical threshold signatures," in *Proc. 17th Int. Conf. Theory Appl. Cryptographic Techn.*, Bruges, Belgium, 2000, pp. 207–220.
[31] A. Kate, Y. Huang, and I. Goldberg, "Distributed key generation in the wild," *IACR Cryptol. ePrint Arch.*, vol. 1, 2012, Art. no. 377.
[32] R. Li and A. X. Liu, "Adaptively secure conjunctive query processing over encrypted data for cloud computing," in *Proc. IEEE 33rd Int. Conf. Data Eng.*, San Diego, USA, 2017, pp. 697–708.
[33] Z. Ma, W. Zhang, H. Fang, X. Dong, L. Geng, and N. Yu, "Local geometric distortions resilient watermarking scheme based on symmetry," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4826–4839, Dec. 2021.
[34] E. V. Mangipudi, K. Rao, J. Clark, and A. Kate, "Towards automatically penalizing multimedia breaches (Extended Abstract)," in *Proc. IEEE 4th Eur. Symp. Secur. Privacy Workshops*, Stockholm, Sweden, 2019, pp. 340–346.
[35] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13rd ACM Conf. Comput. Commun. Secur.*, Alexandria, USA, 2006, pp. 89–98.
[36] F. McKeen et al., "Innovative instructions and software model for isolated execution," in *Proc. 2nd Int. Workshop Hardware Architectural Support Secur. Privacy*, Tel-Aviv, Israel, 2013, pp. 1–1.
[37] I. Anati, S. Gueron, S. Johnson, and V. Scarlata, "Innovative technology for CPU based attestation and sealing," in *Proc. 2nd Int. Workshop Hardware Architectural Support Secur. Privacy*, Tel-Aviv, Israel, 2013, pp. 1–7.
[38] Intel, Intel software guard extensions, 2023. [Online]. Available: https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/get-started.html
[39] Y. Xiao, N. Zhang, J. Li, W. Lou, and Y. T. Hou, "PrivacyGuard: Enforcing private data usage control with blockchain and attested off-chain contract execution," in *Proc. 25th Eur. Symp. Res. Comput. Secur.*, Guildford, U.K., 2020, pp. 610–629.
[40] S. Li, K. Xue, D. S. L. Wei, H. Yue, N. Yu, and P. Hong, "SecGrid: A secure and efficient SGX-enabled smart grid system with rich functionalities," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 1318–1330, 2020.
[41] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 3rd ed., Boca Raton, FL, USA: CRC Press, 2021, pp. 1–598.
[42] H. Yang, W. Shin, and J. Lee, "Private information retrieval for secure distributed storage systems," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 12, pp. 2953–2964, Dec. 2018.
[43] C. Rackoff and D. R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," in *Proc. 8th Annu. Int. Cryptol. Conf.*, Santa Barbara, USA, 1991, pp. 433–444.
[44] InterPlanetary file system, 2023. [Online]. Available: https://www.ipfs.com
[45] H. Wu, J. Zhou, J. Tian, and J. Liu, "Robust image forgery detection over online social network shared images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, USA, 2022, pp. 13430–13439.
[46] Y. Zheng, J. Bao, D. Chen, M. Zeng, and F. Wen, "Exploring temporal coherence for more general video face forgery detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, Canada, 2021, pp. 15024–15034.
[47] D. Pointcheval and O. Sanders, "Short randomizable signatures," in *Proc. Cryptographers' Track RSA Conf.*, San Francisco, USA, 2016, pp. 111–126.



Meng Li (Senior Member, IEEE) received the PhD degree in computer science and technology from the School of Computer Science and Technology, Beijing Institute of Technology (BIT), China, in 2019. He is an associate professor and dean assistant with the School of Computer Science and Information Engineering, Hefei University of Technology (HFUT), China. He is also a postdoc researcher with the Department of Mathematics and HIT Center, University of Padua, Italy, where he is with the Security and PRIvacy Through Zeal (SPRITZ) research group led by Prof. Mauro Conti (IEEE Fellow). He was sponsored by China Scholarship Council (CSC) (from September 1, 2017 to August 31, 2018) for joint PhD study supervised by Prof. Xiaodong Lin (IEEE Fellow) in the Broadband Communications Research (BBCR) Lab with University of Waterloo and Wilfrid Laurier University. His research interests include security, privacy, applied cryptography, and vehicular networks. In this area, he has published more than 60 papers in international peer-reviewed transactions, journals, magazines, and conferences, including *IEEE Transactions on Dependable and Secure Computing*, *IEEE/ACM Transactions on Networking*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Services Computing*, *IEEE Transactions on Smart Grid*, *IEEE Transactions on Industrial Informatics*, *IEEE Transactions on Vehicular Technology*, *IEEE Transactions on Network and Service Management*, *IEEE Transactions on Network Science and Engineering*, *IEEE Transactions on Green Communications and Networking*, MobiCom, ICICS, SecureComm, TrustCom, and IPCCC. He is a Senior Member of IEEE. He is an associate editor of *IEEE Transactions on Information Forensics and Security* and *IEEE Transactions on Network and Service Management*.

Yanzhe Shen received the BS degree from the School of Computer Science and Information Engineering, Hefei University of Technology in 2023. Currently, he is working toward the MS degree with the School of Computer Science and Technology, University of Science and Technology. His research interests include security, privacy, applied cryptography, SGX, blockchain, and vehicular networks.

Guixin Ye received the PhD degree from Northwest University, China, in 2019. He is an associate professor with the School of Information Science and Technology, Northwest University. His research interests include software security, software testing, authentication and privacy. He has published more than 20 papers in international peer-reviewed conferences and transactions, including PLDI, CCS, NDSS, PACT, *IEEE Transactions on Information Forensics and Security*, and *ACM Transactions on Transactions on Privacy and Security*.

Jialing He (Member, IEEE) received the MS and PhD degrees from the Beijing Institute of Technology, Beijing, China, in 2018 and 2022, respectively. She is currently a research assistant professor with the College of Computer Science, Chongqing University, Chongqing, China. Her current research interests include differential privacy, user behavior mining, and blockchain.

Xin Zheng received the MS degree from the School of Business, Anhui University, China. Currently, she is an associate section chief with the Anhui Provincial Department of Justice. Her research interests include criminal law, accounting and digital forensics.

Zijian Zhang (Senior Member, IEEE) received the PhD degree from the School of Computer Science and Technology, Beijing Institute of Technology. He is now an associate professor with the School of Cyberspace Science and Technology, Beijing Institute of Technology. He was a visiting scholar with the Computer Science and Engineering Department, State University of New York at Buffalo in 2015. His research interests include design of authentication and key agreement protocol and analysis of entity behavior and preference.

Liehuang Zhu (Senior Member, IEEE) received the MS degree in computer science from Wuhan University, Wuhan, China in 2001, and the PhD degree in computer science from the Beijing Institute of Technology, Beijing, China in 2004. He is a full professor with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China. His research interests include data security and privacy protection, blockchain applications, and AI security. He has authored more than 150 journal and conference papers in these areas. He is an associate editor of *IEEE Transactions on Vehicular Technology*, *IEEE Network*, and *IEEE Internet of Things Journal*. He was a guest editor of special issue of *IEEE Wireless Communications* and *IEEE Transactions on Industrial Informatics*. He has served as program co-chair of MSN 2017, IWWS 2018, and INTRUST 2014. He received the Best Paper Award at IEEE/ACM IWQoS 2017, IEEE TrustCom 2018, and IEEE IPCCC 2014.

Mauro Conti (Fellow, IEEE) received the PhD degree from the Sapienza University of Rome, Italy, in 2009. After his PhD, he was a postdoc researcher with Vrije Universiteit Amsterdam, The Netherlands. He is a full professor with the University of Padua, Italy. He is also affiliated with TU Delft and University of Washington, Seattle. In 2011 he joined as assistant professor with the University of Padua, where he became associate professor in 2015, and full professor in 2018. He has been visiting researcher with 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 400 papers in topmost international peer-reviewed journals and conferences. He is editor-in-chief for *IEEE Transactions on Information Forensics and Security*, area editor-in-chief for *IEEE Communications Surveys and Tutorials*, and has been associate editor for several journals, including *IEEE Communications Surveys and Tutorials*, *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, CANS 2021, and general chair for SecureComm 2012, SACMAT 2013, NSS 2021 and ACNS 2022. He is senior member of the ACM, and fellow of the Young Academy of Europe.