Eunomia

Anonymous and Secure Vehicular Digital Forensics based on Blockchain

Li, Meng; Chen, Yifei ; Lal, C.; Conti, M.; Alazab, Mamoun ; Hu, Donghui

DOI
10.1109/TDSC.2021.3130583

Publication date
2023

Document Version
Final published version

Published in
IEEE Transactions on Dependable and Secure Computing

Eunomia: Anonymous and Secure Vehicular Digital Forensics Based on Blockchain

Meng Li, *Member, IEEE*, Yifei Chen, *Student Member, IEEE*, Chhagan Lal, *Member, IEEE*, Mauro Conti, *Fellow, IEEE*, Mamoun Alazab, *Senior Member, IEEE*, and Donghui Hu, *Member, IEEE*

Abstract—Vehicular Digital Forensics (VDF) is essential to enable liability cognizance of accidents and fight against crimes. Ensuring the authority to timely gather, analyze, and trace data promotes vehicular investigations. However, adversaries crave the identity of the data provider/user, damage the evidence, violate evidence jurisdiction, and leak evidence. Therefore, protecting privacy and evidence accountability while guaranteeing access control and traceability in VDF is no easy task. To address the above-mentioned issues, we propose Eunomia: an anonymous and secure VDF scheme based on blockchain. It preserves privacy with decentralized anonymous credentials without trusted third parties. Vehicular data and evidence are uploaded by data providers to the blockchain and stored in distributed data storage. Each investigation is modeled as a finite state machine with state transitions being executed by smart contracts. Eunomia achieves fine-grained evidence access control via ciphertext-policy attribute-based encryption and Bulletproofs. A user must hold specific attributes and a temporary-and-unexpired token/warrant to retrieve data from the blockchain. Finally, a secret key is embedded into data to trace the traitor if any evidence breach happens. We use a formal analysis to demonstrate the strong privacy and security properties of Eunomia. Moreover, we build a prototype in a WiFi-based Ethereum test network to evaluate its performance.

Index Terms—Vehicular networks, digital forensics, privacy, security, blockchain

1 Introduction

Smart vehicles are often equipped with functionally rich sensors, control units, and communication units. They perceive surrounding environments, process local data, and interact with nearby vehicles, Road-Side Units (RSUs), and service providers. With the abundant vehicular data, drivers will be provided with more real-time services [1], [2], [3], and the traffic department will be channeling more intelligence in road construction planning. Unfortunately, traffic accidents are happening worldwide, and vehicles are involved in crimes or used as a means of crime in terrorist attacks. For these accidents and crimes, digital forensics customized for vehicular networks can be performed to collect vehicular data and feed investigations [4], which is called Vehicular Digital Forensics (VDF) [5]. It consists of collection, examination, analysis, and report [6]. *Data providers*, e.g., vehicles and pedestrians, upload their sensed traffic data to the police department or the traffic management center. *Data users*, e.g., a traffic police investigator, request data about traffic offenders, vehicles, and accident scenes to complete liability cognizance of traffic accidents. A criminal police investigator has to request a court-issued warrant [7] to request data relevant to suspects, vehicles, or crime scenes, identify malicious activities to conduct a vehicular crime investigation.

Despite offering a convenient and effective way of managing forensic data, several threats raise high privacy and security concerns [8], [9], [10], [11]. First, some witnesses choose not to provide evidence to an investigator if they fear intentional retaliation from a hostile suspect. Second, the uploaded data in plaintext may be exposed to an outlaw, which warns him to change criminal plans. The contents of the token/warrant may be exposed to a malicious crime police investigator who sabotages the investigation covertly. Third, a malicious data user may exceed his authority to request more data uncorrelated to his case from the database. He may also use an expired token/warrant to enforce invalid powers. Moreover, a malicious insider may modify the data in the database, and a malicious data provider may claim a loss of data, which undermines the digital forensics chain of custody [8]. Last, some malicious data users who have retrieved data from the database may accidentally leak the data or deliberately sell it to the black market for money [9]. Therefore, it is

- *Meng Li, Yifei Chen, and Donghui Hu are with the Key Laboratory of Knowledge Engineering with Big Data, Ministry of Education, School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, Anhui 230009, China, and also with Anhui Province Key Laboratory of Industry Safety and Emergency Technology, Intelligent Interconnected Systems Laboratory of Anhui Province, Hefei University of Technology, Hefei, Anhui 230009, China. E-mail: {mengli, hudh}@hfut.edu.cn, yifeichen@mail.hfut.edu.cn.*
- *Chhagan Lal is with Delft University of Technology, Netherland. E-mail: c.lal@tudelft.nl.*
- *Mauro Conti is with the Mauro Conti is with the Department of Mathematics, University of Padua, 35131 Padua, Italy. E-mail: conti@math.unipd.it.*
- *Mamoun Alazab is with the College of Engineering, IT and Environment, Charles Darwin University, Casuarina, NT 0810, Australia. E-mail: alazab.m@ieee.org.*

Manuscript received 8 February 2021; revised 24 October 2021; accepted 22 November 2021. Date of publication 25 November 2021; date of current version 16 January 2023.
This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 62002094, the Anhui Provincial Natural Science Foundation under Grant 2008085MF196, Anhui Science and Technology Key Special Program under Grant 201903a05020016, and the National Natural Science Foundation of China (NSFC) under Grant U1836102. It is partially supported by EU LOCARD Project under Grant H2020-SU-SEC-2018-832735.
(Corresponding author: Donghui Hu.)
Digital Object Identifier no. 10.1109/TDSC.2021.3130583

nontrivial to solve these privacy and security challenges in VDF simultaneously.

We note that while some related schemes have been proposed [6], [12], there are still many issues to be explored, which lead to our *three motivations* behind this work. First, although finite state machine following the verification-then-forwarding model could model a VDF process, it would be more realistic and practical to consider the state transitions of investigations rather than warrants [12]. A warrant can expire if the statute of limitations is up or some warrants should be executed before a certain date [13]. Second, the intentional or unintentional data breach was not considered either. Traitor tracing is a crucial remedial measure for digital forensics, given that data leakage [14] causes catastrophic consequences to investigations. Third, most VDF works do not formally define security and privacy, let alone formal proof of security and privacy.

The aforementioned problems and motivations lead to the *three technical challenges* in designing anonymous and secure VDF. *Challenge I*: from the privacy perspective, how to protect the privacy of both data providers and data users who have different privacy requirements in an open VDF system based on blockchain? In none-blockchain-based VDF systems, user privacy can be guaranteed if we secure the communication channel between the user and the server while preventing the server from being breached. However, we adopt the blockchain to infuse VDF with unforgeability and verifiability, which makes the transmitted messages transparent [15]. Considering that data providers and data users have different data to protect, it is challenging to provide privacy. *Challenge II*: from the security perspective, how to defend an anonymous VDF system from data leakage attacks? Privacy goals can be achieved by adaptively leveraging privacy-preserving techniques. However, in VDF, there may be inside adversaries who leak the data, which incurs severe consequences to the investigations and the justice system. This requires traitor tracing that is challenging to do in a VDF scheme providing anonymity as well as blockchain-based systems for non-VDF applications. Meanwhile, we need to formally define and prove privacy and security for anonymous and secure VDF. Such a VDF scheme is also a protocol including entities and functions. Different from cryptographic protocols, it needs adjustments when we are treating privacy and security. *Challenge III*: from the performance perspective, how to build an anonymous and secure blockchain-based VDF system while maintaining acceptable costs on computation and communication given the energy constraints on smartphones and pertinent devices? We aim to provide several privacy and security goals that require using cryptographic techniques. Instead of arbitrarily stuffing our system with such techniques, we first need to integrate them to keep the system up and running. Furthermore, we should control the costs as low as possible by adjusting parameters, optimizing data structures without sacrificing security.

To address these challenges, we leverage blockchain [16], smart contract [17], and several carefully bridged cryptographic primitives to design a novel VDF scheme Eunomia. It provides anonymity, unlinkability, data confidentiality, authentication, fine-grained access control, accountability, and traceability. We build a consortium blockchain among local authorities to collect evidence from data providers. We design a smart contract that models each vehicular forensics investigation as a finite state machine, putting the investigation procedures under supervision. We frame the key contributions as follows.

- We propose a novel framework for VDF including data providers (e.g., vehicles and pedestrians), data users (e.g., crime police investigator), authorities (e.g., court), a blockchain network, and distributed data storage system. We enhance its security model by considering data breaches from malicious data users. In this model, the honest-but-curious assumption is adopted for most entities and a small part of investigators can launch data tampering, unauthorized data accessing, and data leakage attacks. The VDF investigation is modeled as a finite-state machine in smart contracts.
- We propose a blockchain-based VDF scheme named Eunomia with strong privacy and security guarantees. Decentralized anonymous credentials [18] are utilized to preserve anonymity and unlinkability (data provider privacy and witness privacy [19]). Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [20] and Bulletproofs [21] are jointly used to protect the data confidentiality, realize fine-grained access control, and verify token/warrant validity. A consortium blockchain is constructed to record data-related transactions and guarantee accountability. The uploaded data is stored in the Ali distributed cloud storage system [22]. Next, a committed user oblivious transfer protocol is designed to achieve traceable data requests, i.e., the data requester's secret key is watermarked into data in case of data leakage.
- We give formal definitions of privacy and security, and then give formal analysis to prove these properties. We construct a prototype via a WiFi-based Ethereum test network with desktop/laptop computers as blockchain nodes and smartphones as data providers/users to demonstrate its feasibility and evaluate the computational costs, communication overhead, network delay, monetary cost, power consumption, and scalability.

The remainder of this paper is organized as follows. We review some related work in Section 2. Section 3 introduces some preliminaries. Section 4 formalizes the problem. In Section 5, we elaborate Eunomia, followed by the privacy and security analysis in Section 6 and performance evaluation in 7, respectively. Finally, we provide some discussions in Section 8 and conclude the paper in Section 9.

2 Related Work

2.1 Digital Forensics

A digital forensics investigation procedure includes some phases, including evidence acquisition and duplication, evidence analysis, and result presentation [23]. An investigator collects evidence from a crime scene or witnesses with professional tools. The evidence is sent to an analyst for systematical analysis. Based on the analysis report and case reasoning, the investigator draws a case result. Digital Forensics has transformed from a stealth to an essential

component in solving cases and developments in forensic tools and research have been very triumphant [24]. Frankle et. al [8] has introduced secrecy and accountability in electronic surveillance. Based on cryptographic commitments, secure multiparty computation, zero-knowledge arguments, and an append-only ledger, they provide a wide range of choices for improving accountability with necessary secrecy to the US court system.

2.2 Vehicular Digital Forensics

Digital forensics has been integrated with vehicular networks and some VDF works are proposed in the literature. Empowered by multiple Electronic Control Units (ECUs), digital forensics investigations can feed on the evidence collected by the ECUs. There are not many centralized schemes or protocol on VDF and most of them are far from maturity. Mansor *et al.* [25] proposed a secure and privacy-preserving scheme to achieve a forensics data collection and storage process in automotive systems. Feng *et al.* [26] presented a straightforward VDF scheme which only addressed data confidentiality. Huang et. al [27] formalized the vehicular fog computing architecture and presented a typical use case. They only discussed some security and forensic challenges, and potential solutions. Le-Khac *et al.* [28] discussed some challenges in vehicular forensics, conducted a case study on data acquisition and analysis on a Volkswagen car, and presented potential forensic solutions via hardware and software.

2.3 Blockchain-Based Vehicular Digital Forensics

Although VDF and blockchain attracted attention in the past a few years, there are not many blockchain-based VDF studies. Still, they are gaining increasing research heat. Cebe *et al.* [6] stated that connected and smart vehicles would provide valuable data to different stakeholders, such as maintenance companies, vehicle manufacturers, drivers, and insurance companies. They presented a framework for managing vehicular data, which combines the vehicular public key infrastructure (i.e., pseudonym certificates) with a permissioned blockchain to establish membership and protect privacy. Next, they proposed a fragmented ledger to preserve detailed vehicular data, e.g., diagnosis records and maintenance reports. However, the authors just proposed the framework design and no implementation or evaluation is done.

Dai *et al.* [29] presented an indirect reciprocity security framework with a scalar reputation for each OBU to assess the dangerous level to the VANET. A blockchain is utilized to protect information from being tampered with and record the behaviors of other OBUs. They also presented a reinforcement learning based action selection strategy for an OBU to select a reliable relay OBU and decide whether to follow the request of a source OBU.

Li *et al.* [12] presented a blockchain-based scheme BB-VDF consisting of accountable protocols and privacy preservation methods. They modeled the forensics procedures as a finite state machine and realized the state transition via a smart contract. They used Distributed Key Generation (DKG) in (t, n)-threshold cryptosystem to require a law enforcement agency to have at least $t + 1$ decryption shares to decrypt the requested data. They also designed a distributed key-policy attribute based encryption scheme to realize the secure fine-grained access control. However, this scheme does not address the anonymity, unlinkability or traceability.

The advantages of the proposed scheme compared with the existing blockchain-based vehicular digital forensic schemes are threefold. First, we model each investigation as a finite state machine to keep our design true to reality. Second, we have utilized zero-knowledge proofs to privately verify the validity of a token/warrant. Third, we can trace the traitor who leaks requested data by embedding the traitor's secret key into the requested data.

3 Preliminaries

In this section, we briefly revisit several underlying techniques employed to structure Eunomia.

3.1 Bilinear Map and Mathematical Assumptions

A map e from multiplicative groups $\mathbb{G}_1$ and $\mathbb{G}_2$ to a multiplicative group $\mathbb{G}_T$, is a bilinear map if for all $a, b \in \mathbb{Z}, x \in \mathbb{G}_1, y \in \mathbb{G}_2$, it satisfies that $e(x^a, y^b) = e(x, y)^{ab}$. e is non-degenerate if $e(x, y) = 1$ implies that either $x = 1$ or $y = 1$. e is asymmetric if no efficiently computable homomorphism exists between $\mathbb{G}_1$ and $\mathbb{G}_2$.

Strong RSA assumption. For a randomly generated RSA modulus n and a random element $y \in \mathbb{Z}_n^*$, it is difficult to find a probabilistic polynomial-time algorithm that calculates $x \in \mathbb{Z}_n^*$ and integer $i > 1$ satisfying $x^i = y \bmod n$.

Discrete logarithm assumption. Let G be a cyclic group with a generator g. Given $y \in G$, it is difficult to calculate x satisfying $y = g^x$.

Decisional linear assumption. An asymmetric pairing group generator GGen satisfies the decisional linear assumption (DLIN) if for all Probabilistic Polynomial-Time (PPT) adversaries $\mathcal{A}$, $\mathrm{Adv}^{\mathcal{A}}_{\mathrm{DLIN}} = |\Pr[\mathcal{A}(1^k, pp, D, T_0)] - \Pr[\mathcal{A}(1^k, pp, D, T_1)]|$ is negligible where $pp := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h) \leftarrow \mathrm{GGen}(1^k)$, $a, b \leftarrow_R \mathbb{Z}_p^*$; $c, d, f \leftarrow_R \mathbb{Z}$; $D := (g^a, g^b, h^a, h^b,\ g^{ac}, g^{bd}, h^{ac}, h^{bd})$; $T_0 := (g^{c+d}, h^{c+d})$; $T_1 := (g^f, h^f)$.

3.2 Zero-Knowledge Proof

In a zero-knowledge proof (ZKP) protocol [30], a prover proves a statement to a verifier without revealing anything about the statement other than that it is true. Normally, the prover and the verifier have to conduct several interactions for the proving process to complete. But this process can be converted into noninteractive proofs by applying the Fiat-Shamir heuristic [31]. For example, we denote $\mathrm{NIZKPoK}\{(x) : y = g^x\}$ as s a non-interactive zero-knowledge proof of knowledge of the elements x satisfying $y = g^x$. The values in the $()'$ are the knowledge that the prover needs to prove, and the other values are known to the verifier. The $\mathrm{ZKSoK}[m]\{(x) : y = g^x\}$ indicates a signature of knowledge on message m.

3.3 Pedersen Commitment

A commitment scheme enables a user to bind himself to a chosen value x without revealing x to the commitment receiver. This commitment c to x ensures that the user cannot change his choice and ensures that the commitment receiver does not learn anything about x. These two attributes are

called binding and hiding. In Pedersen commitment [32], a group G of prime order q and generators $(g_0, g_1, ..., g_m)$ are given. To commit to values $(x_1, ..., x_n) \in \mathbb{Z}_q^n$, the user randomly chooses $r \in \mathbb{Z}_q$ and computes $c = \text{Com}(x_1, ..., x_m; r) = g_0^r \Pi_{i=1}^n g_i^{x_i}$.

3.4 Ciphertext-Policy Attribute-Based Encryption

A Ciphertext-Policy Attribute-Based Encryption (CP-ABE) scheme [33] consists of four algorithms as follows:

Setup (1^k): given a security parameter k, outputs a public key pk and a master secret key msk.

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

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

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

The motivation of using CP-ABE is to realize access control of evidence toward different entities. Given that different entities have different authorities and access levels, it is reasonable to grant them different access to evidence.

3.5 Blockchain and Smart Contract

As one of the underlying techniques in Bitcoin [16], blockchain is essentially a public ledger which records transactions among users. The transactions are packed into separate blocks by a group of blockchain nodes (miners) using a certain consensus mechanism (e.g., proof of work (PoW), proof of stake (PoS), and practical Byzantine fault tolerance (PBFT)), and the blocks are linked into a chain by their cryptographic hashes. These nodes participate in the mining activities to compete for some rewards such as financial incentives. Consortium blockchain is a specific blockchain maintained by a group of authorized entities. Only the parties in the blockchain system 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. In a consortium blockchain, the consensus process is controlled by the authorized entities. Some famous consortium blockchains are hyperledger fabric, Corda, Coco, and EEA. The motivation of using blockchain is to record the evidence-related activities in a tamperproof and veritable ledger. In this way, we can monitor and track them in a reliable way.

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. Smart contracts enable general computations on the blockchain. For example, Ethereum is the first Turing-complete decentralized smart contract system.

4 Problem Statement

In this section, we formalize the architecture of Eunomia and lay out its security assumptions in Section IV.A and threat model in Section IV.B. Then we describe its privacy, security, and efficiency objectives in Section IV.C.

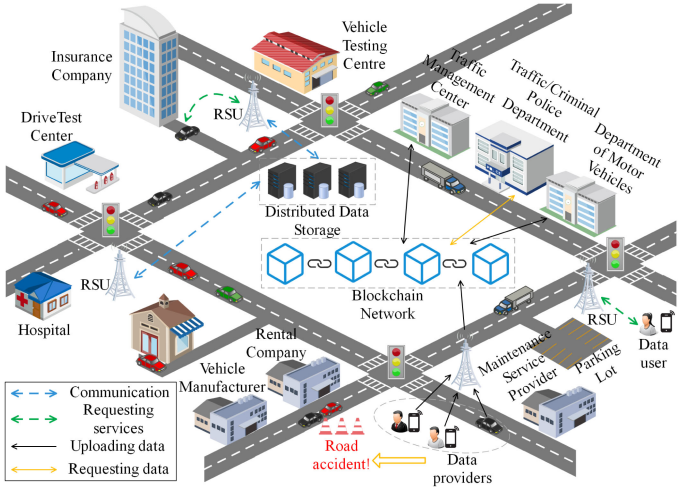


Fig. 1. System model of Eunomia.

4.1 System Model

The Eunomia system model consists of five types of entities: Data Providers (DP), Data Users (DU), Authorities (AU), Consortium Blockchain (CB), and Distributed Data Storage System (DDSS). It is displayed in Fig. 1 and key notations are explained in Table 1.

There are two types of data, namely *in-vehicle data* and *out-vehicle data*. The former refers to vehicle status (e.g., speed, location, steering wheel angle) collected by an Event Data Recorder (EDR), driver status collected via wearables, and maintenance history. The latter refers to the vehicle data (e.g., brake, tyre, in-vehicle heat) and road data (e.g., braking trace, traffic light status, road conditions, and weather) recorded in photos, audio, and videos which are collected by smartphones and monitors from nearby vehicles, pedestrians, and RSUs. Besides, there are three types of data formats, namely *photo*, *audio*, and *short video*. One of our motivations for this work is to allow vehicular users including drivers, pedestrians, and in-vehicle devices (especially the monitor) to upload fresh on-road data. It is also convenient for the vehicular users to take a photo and record audio with their smartphones or the in-vehicle devices to capture a video. Therefore, we choose photo, audio, and short video as the three exemplary types. The textual or numeric vehicle data are supported in our settings via transforming them into a photo. If the data is traffic related, it will be uploaded to the Traffic Police Department (TPD). If the data is crime related, it will be uploaded to the Crime Police Department (CPD).

We give a representative example to present all the steps (flow) of Eunomia in a simple way. Say Alice has witnessed a truck ramming into a jewellery store and the truck driver fled away with some pricey necklaces. She takes a picture of the driver with her smartphone. Then she opens the Eunomia app to submit an upload transaction to the blockchain and sends an encrypted picture to the storage system. An investigator Bob leads the investigation on the robbery. He uses the Eunomia app to submit a request transaction to seek useful evidence from the blockchain. After the request is verified by a blockchain node, a download link is returned to Bob. Then Bob submits the link to the police department and interacts with it to embed his private key

TABLE 1
Key Notations of Eunomia

Notation	Definition
dp, du, au	Data provider, data user, authority
cb	Consortium blockchain
ds	Distributed data storage system
cpi, co	Crime police investigator, court
tpi	Traffic police investigator
p_1, q_1, p_2, q_2; N; s	Prime; product of p_1 and q_1; seed
g_1, g_2, $h_0, ..., h_n$	Group generator
$\mathbb{G}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$; e	Group; bilinear map
p, pk; msk	Prime order, public key; master secret key
a_1, a_2, b_1, b_2, b_3, c_1, c_2	Random number
A_1, A_2, B_1, B_2	Part of public key
H_1, H_2, H_3	Hash functions
puk^{au}, prk^{au}	au's public key and secret key
m, att	Number of attributes, attribute set
$att_i^0, ..., att_l^{m-1}$; aux	Attribute; auxiliary data
sk, pi	Master secret, pseudo-identity
sk, cr	Secret key, credential
π; Tx	Signature of knowledge; transaction
ac	Anonymous credential
$\mathbb{C}$	Set of credentials
$\mathbb{A}$	Set of investigation attributes
$\overline{prk}_j$, $\overline{puk}_j$	du's master private and public key
$\bar{r}_1, \bar{r}$	Random numbers
sk^A, to, $wa\ d$, md	Attribute key, token, warrant
d, md	Data, metadata
$\mathbb{S}$, $\mathbb{U}$	Access structure, universe of attributes
$\mathbf{M}$, f	Matrix, mapping function
n_1, n_2; n_3	Matrix size; number of credentials
u_1, u_2	Random number
Ed; $ct_i^0, ..., ct_i^{n_1}, ct_i'$	Encrypted data
π'	Zero-knowledge proof of knowledge
HV, X, π''	Hash value, case number, range proof
L, σ, db	Download link, length of $\overline{prk}_j$, data block
db^0, db^1; C, R	Watermarked data block; Commitment
Ed', C	Re-encrypted data, ciphertext

into the requested data. The encrypted data are returned to Bob. If Bob holds enough access attributes, he will be able to decrypt the ciphertext. Next, Bob conducts analysis on the data and prepares a digital report. If the data is released by Bob, his identity can be revealed based on the watermarks on the data, i.e., his secret key.

Data providers are the entities which upload data for public interests or economic incentives in some services, such as Amazon Mechanic Turk. DPs send the encrypted data to the DDSS and upload a data uploading transaction including the hash of encrypted data as well as the metadata (e.g., data type, data format, date, and timestamp) to the CB. Main data provides are vehicles and pedestrians. Each vehicle is equipped with a GPS module, an electronic control unit, and an On-Board Units (OBUs). A pedestrian is holding a smartphone with sensing and communicating capabilities.

Data users are the entities which request data needed for an investigation. Before a competent DU requests data, she/he has to obtain a set of investigation attributes and a valid token/warrant from an authority. The attributes include cognitive abilities, knowledge base, life experience, work orientation, communication skills, professional demeanor, and approach to life [34]. Then the DU sends a data accessing transaction to the CB to request data. Before the data is returned to the DU, the DU and an authority execute an interactive protocol to embed the DU's identifier, i.e., private key, into the plaintext data, ensuring the potential data leakage traceable. Only the authorized police investigators can retrieve the corresponding data from the storage system. By authorized police investigators, we mean the ones who possess a valid token/warrant and a corresponding set of attributes. Main DUs are Traffic Police Investigator (TPI) and Traffic Analyst (TA) in the TPD, and Crime Police Investigator (CPI) and Legal Medical Expert (LME) in the CPI.

Authorities are the entities responsible for initializing the whole VDF system. AUs include the court, CPD, TPD, and some Governmental Departments (GDs) which have administrative powers in traffic governance ministry and security ministry. The CPD obtains the uploaded data by requesting the data from the storage system. It is responsible for making the leakage of requested data traceable by executing an interactive protocol with DUs. The protocol embeds the DU's private key in the requested data. We assume the execution environment of the AUs is run in a trusted and secure hardware enclave such that adversaries cannot acquire the contents of stored data. The court is a sacred institution that approves or denies a CPI's warrant request by a judge according to the legal provision. It also holds trials, preserves court data, and tallies votes, which is not the focus of this paper. The authorities stay at a higher level of the system, and they maintain the blockchain together.

A consortium blockchain is a blockchain that is maintained by several Blockchain Nodes (BNs), including governmental institutions, such as CPD. The blockchain records three types of data-related transactions: (1) data uploading transaction, (2) data accessing transaction, and (3) data retrieving transaction. In (1), the hash of encrypted data is attached. The CB also keeps track of the state of each investigation. There are eight states, namely *warrant request*, *investigation initiate*, *data request*, *data retrieval*, *data analysis*, *result report*, *investigation closure*, and *completed* as depicted in Fig. 2. The blockchain enables the whole system to supervise the transparency and legitimacy of the investigation, making it auditable.

Distributed data storage system is the storage system that stores encrypted data. When an investigator requests some data, she/he has to possess a valid warrant and a decryption key. After data access request is received, the CB retrieves corresponding data from the DDSS and returns a download link to the investigator. As for searching for data, we support two approaches: (1) metadata, and (2) dynamic secure searchable encryption (DSSE) [35]. In (1), data providers have to upload some metadata as a tag to describe the detailed data, and the investigator can choose her/his metadata to search for data. In (2), we adopt the DSSE that achieves forward and backward privacy to store and search data securely.

4.2 Threat Model

We adopt the honest-but-curious assumption for most internal entities which strictly follow the protocols but try to learn the privacy of DPs, the contents of the uploaded data and the contents of the warrant. A part of insiders may be crime

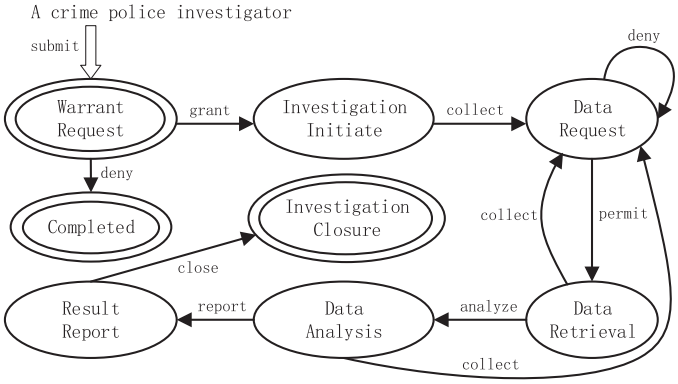


Fig. 2. The state machine model for VDF.

complice or lured by money temptation. These malicious entities, i.e., investigators and analysts, can launch data falsification, unauthorized access, and data leakage attacks.

Data falsification attack refers to malicious entities' tampering with the data that is already uploaded to the CB and stores in the DDSS. *Unauthorized access attack* refers to malicious entities' attempting to request data from the blockchain while the data is beyond their access jurisdiction. *Data leakage* attack refers to malicious entities' accidentally leaking the requested data to the public, i.e., deliberately leaking the data to criminals, mafia organizations, and the black market.

The data falsification attack destroys the data integrity by falsifying the uploaded data in the storage system. The unauthorized access attack breaks the access control by requesting and accessing data outside the adversary's jurisdiction. The data leakage attack damages the access control and indicates a traitor by revealing the requested data to an unqualified entity or even a malicious criminal.

External adversaries eavesdrop on the communication channel and launch several typical attacks, i.e., impersonation attack, replay attack, in an attempt to violate privacy and sabotage the system. They may compromise an investigator to tamper with or access the uploaded data. They can also compromise an analyst to forge an analysis result.

4.3 Design Objectives

To resist the threats mentioned above, we list the following privacy and security goals. We also provide some definitions using cryptographic games [36] in the Appendix.

Privacy contains anonymity and unlinkability.

- Anonymity. The identity of DPs/DUs should be anonymous when DPs/DUs are uploading/accessing data to/from CB.
- Unlinkability. Two data transactions sent by the same DP/DU are indistinguishable such that the identity and data content/request from the same DP/DU in data uploading/accessing cannot be linked.

Security contains confidentiality, fine-grained access control, accountability, and traceability.

- Data confidentiality. The contents of the uploaded data should be protected from unqualified entities.
- Authentication. The identity of DPs and DUs should be authenticated before their data transactions are accepted to screen illegal entities.
- Fine-grained access control. The system must verify the attributes of DUs and the validity of warrants to rule out unauthorized accesses and expired accesses.
- Accountability. The system should protect the digital forensics chain of custody, i.e., integrity and auditability of data, so the data is hard to tamper with and easy to be audited.
- Traceability. The system must be able to trace the source who leaks her/his previously requested data when a lawful enforcement agency is aware of the data breach and spots the corresponding *plaintext data*.

5 Proposed Scheme

In this section, we present the details of the proposed Eunomia. Subsection V.A describes the overview of Eunomia and blockchain-based VDF state machine. The following subsections present the concrete scheme.

5.1 Overview

There are seven phases in Eunomia, namely system Initialization, entity registration, data uploading, data accessing, data retrieval, data analysis and result reporting, and traitor tracing. We use an example to show the general process of VDF. First, a pedestrian photographed a "hit and run" with a smartphone. Given an Eunomia app on her/his smartphone, the pedestrian uploads the photos to the Eunomia system, i.e., sending the hash value of the photos to the blockchain while sending encrypted photos to the DDSS. Meanwhile, a vehicle nearby also witnesses the whole accident and uploads a short video to the Eunomia system. Later, a CPI submits a warrant request to conduct an investigation and submits the request to the court. The court issues a valid warrant to the CPI and then the investigator holds a set of attributes related to the investigation. To acquire the pedestrian and vehicular data, the CPI uses the warrant to generate a data request and submits it to the blockchain. A blockchain node verifies the validation of the request, searches corresponding data, and sends back a download link to the CPI. The CPI downloads a part of the data from DDSS and decrypts an identifier of an AU. Next, the CPI interacts with the AU to obtain the requested data. As for the data analysis and reporting, which are not the focus of this work, we follow the idea of [6].

We portray a state machine to illustrate VDF in smart contract as shown in Fig. 2, which has eight states, namely *warrant request*, *investigation initiate*, *data request*, *data retrieval*, *data analysis*, *result report*, *investigation closure*, and *completed*. The state machine describes the life cycle of an investigation. Each node in the figure denotes a global state of the investigation. Different actions trigger the state transition. Each state transition is attached with a digital signature in a transaction and verified by multiple blockchain nodes. For example, a CPI submits a warrant request to the court and activates the initial state *warrant request*. After the court grants the request, the state transits to *investigation initiate*; otherwise, it goes to *completed*. Once an investigation is opened, the CPI collects some data and evidence needed to establish the elements of criminal activity. The state transits to *data request*. If the request is permitted by a blockchain node and an AU, the state transits to *data retrieval*. The CPI

can continue to collect data until she/he obtains all the data or the gathered data is sufficient. When the data is analyzed and reported, the state transits to *data analysis* and *result report*, successively. In *data analysis*, the TA or LME conducts professional analysis on the data to complete a result report. The goal of the result report is to present a conclusion to an investigation, such as who is the victim, who is the criminal, how the accident/crime happened, and what are the total damages. After the investigation is closed, the state finally transits to *investigation closure*.

5.2 System Initialization

Upon a security parameter 1^k, the authorities $\{au_i\}_{i=1}^{N_1}$ first generate parameters for the decentralized anonymous credential module. They choose two primes p_1 and q_1, compute $N = p_1q_1$, and select a seed $s \in QR_N$ with $s \neq 1$; and generate primes p_2 and q_2 satisfying $p_2 = q_2 2^r + 1$ for some $r \geq 1$. Let $\mathbb{Z}$ be an order-q subgroup of $\mathbb{Z}_{q_2}^*$ and they randomly choose generators $h_0, h_1, ..., h_n$ satisfying $\mathbb{G} = \langle h_0 \rangle = \langle h_1 \rangle = ... = \langle h_n \rangle$.

Second, $\{au_i\}_{i=1}^{N_1}$ generate parameters for the attribute-based encryption module. They choose three groups $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ of prime order $p = \Theta(k)$ with a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Two generators g_1 and g_2 are selected for $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. They choose $a_1, a_2 \leftarrow_R \mathbb{Z}_p$ and $b_1, b_2, b_3 \leftarrow_R \mathbb{Z}_p$, and set $(g_2, A_1 = g_2^{a_1}, A_2 = g_2^{a_2}, B_1 = e(g_1, g_2)^{a_1 b_1 + b_3}, B_2 = e(g_1, g_2)^{a_2 b_2 + b_3})$ as the public key pk. They choose $c_1, c_2 \leftarrow_R \mathbb{Z}_p^*$ and set $(g_1, g_2, a_1, a_2, c_1, c_2, g_1^{b_1}, g_1^{b_2}, g_1^{b_3})$ as master secret key msk. Three hash functions are chosen as $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \{0,1\}^* \rightarrow \{0,1\}^{l_2}$, $H_3 : \mathbb{G}_1 \rightarrow \{0,1\}^{l_3}$. We choose the encryption key size according to an application's requirement on the security level. The security level increases with the key size.

Third, $\{au_i\}_{i=1}^{N_1}$ determine the parameters for consortium blockchain cb (i.e., consensus mechanism, block creation time, authorized nodes, node account deposit, network ID of cb) and initialize the distributed data storage system ds (e.g., create RAM users, and generate accessID and accessKey).

Last, each au_i generates a pair of public key and private key (puk_i^{au}, prk_i^{au}). In summary, the system public parameters are $pp. = (N, s, p_2, q_2, h_0, ..., h_n, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \ e, g_1, g_2, pk, \{puk_i^{au}\})$.

5.3 Entity Registration

A data provider dp_i registers as follows.

- A data provider dp_i holding an attribute set $att_i = (att_i^0, ..., att_i^{m-1})$ (where each item stands for one attribute, e.g., age, height, or sex) and auxiliary data aux_i justifying the credential issue.
- dp_i selects a random $sk_i \in \mathbb{Z}_q$ as the master secret; randomly selects a $r_i \in \mathbb{Z}_q$, computes a pseudo-identity $pi_i = h_0^{r_i} h_1^{sk_i}$, sets secret key $sk_{pi_i} = r_i$.
- dp_i selects a random $r' \in \mathbb{Z}_q$, computes a credential $cr_i = h_0^{r_i} h_1^{sk_i} \prod_{o=0}^{m} h_{o+2}^{att_i^o}$, computes a signature of knowledge on aux_i that pi_i and cr_i belong to the same sk_i: $\pi_j = \text{ZKSoK}[aux_j] \ \{(sk_i, r_i', r_i) : cr_i = h_0^{r_i} h_1^{sk_i} \prod_{o=0}^{m} h_{o+2}^{att_i^o} \wedge pi_i = h_0^{r_i} \ h_1^{sk_i}\}$. The zero-knowledge proof is used to prove that the pseudo-identity and the credential is linked to the master key.
- dp_i submits an entity registration transaction to the cb

$$Tx_1 = [\text{"Register"}, au, ac_i, date, ts]_{prk_i}, \quad (1)$$

where $ac_i = (cr_i, \pi_i, att_i, pi_i, aux_i)$ is dp_i's anonymous credential and ts is timestamp.
- The blockchain nodes accept the credential if the proof verifies successfully.

A data user du_j registers as follows:

- A data user du_j generates a similar $ac_j = (cr_j, \pi_j, att_j, aux_j, pi_j)$ and registers with an au with which she/he is affiliated to obtain a set of investigation attributes $\mathbb{A}_j$ related to the investigation, and a key pair $(\overline{prk_j}, \overline{puk_j})$. Recall the example in Section IV. A, Bob's private key is randomly generated by using the command "Geth account new" when Bob is using a computer. When Bob has an Android smartphone to interact with the system, the private key is randomly generated via "Mobile Account Management" of the mobile Geth library[1].
- au randomly chooses $\bar{r}_1, \bar{r}_2 \leftarrow_R \mathbb{Z}_p$, computes $sk_j^1 := (sk_j^{11}, sk_j^{12}, sk_j^{13}) = (g_2^{c_1\bar{r}_1}, g_2^{c_2\bar{r}_2}, g_2^{\bar{r}_1+\bar{r}_2})$.
- For all the attribute $x \in \mathbb{A}_j$ and $z = 1, 2$, au randomly chooses $r_x \leftarrow_R \mathbb{Z}_p$, and computes $sk_j^{x,z} = H_1(x1z)^{\frac{c_1\bar{r}_1}{a_z}} \cdot H_1(x2z)^{\frac{c_2\bar{r}_2}{a_z}} \cdot H_1(x3z)^{\frac{\bar{r}_1+\bar{r}_2}{a_z}} \cdot g_1^{\frac{r_x}{a_z}}$, sets $sk_j^x := (sk_j^{x,1}, sk_j^{x,2}, sk_j^{x,3} = g_1^{-r_x})$, computes $sk_j^{z'} := g_1^{b_z} \cdot H_1(011z)^{\frac{c_1\bar{r}_1}{a_z}} \cdot H_1(012z)^{\frac{c_2\bar{r}_2}{a_z}} \cdot H_1(013z)^{\frac{\bar{r}_1+\bar{r}_2}{a_z}} \cdot g_1^{\frac{r'}{a_z}}$ for $r' \leftarrow_R \mathbb{Z}_p$ and $z' = 1, 2$, sets $sk_j^3 = (sk_j^{1'}, sk_j^{2'}, sk_j^{3'} = g_1^{b_3} \cdot g_1^{-r'})$, and returns $(sk_j^1, \{sk_j^x\}_{x \in \mathbb{A}_j}, sk_j^3)$ to du_j as the attribute key sk_j^A.
- au also submits a warrant grant transaction to the cb to authorize du_j

$$Tx_2 = [\text{"Grant"}, au, ac_i, date, ts]_{prk^{au}}, \quad (2)$$

where we assume that the CPD and TPD have all the attribute keys that are used in data retrieval later.
- Specifically, a traffic/crime police investigator tpi_j/cpi_j registers with the TPD/court and obtains cd_j, $\mathbb{A}_j$, a token/warrant to_j/wa_j, and sk_j^A. The token and warrant are limited to a range. For example, co has permitted cpi to conduct investigation from the date of warrant request to May 31, 2021 which transforms $wa_j \in [0, 210531]$. Note that we use the cpi as the data user in the following subsections for a specific description.

5.4 Data Uploading

The data provider first preprocesses the data to be uploaded as follows:

- A data provider dp_i has some data d_i and metadata md_i. Recalling the example in Section IV.A, say Alice witnessed the robbery on May 23, 2020 and took a picture of the scene.
- Alice defines an access structure $\mathbb{S}_i$ that is a collection of non-empty subsets of the universe of attributes $\mathbb{U}$. $\mathbb{S}_i$ is also a monotone span program (MSP) [37] that

1. https://geth.ethereum.org/docs/dapp/mobile-accounts

is given by a matrix $\mathbf{M}$ of size $n_1 \times n_2$ over $\mathbb{Z}_p$ and a mapping $f : \{1, ..., n_1\} \rightarrow \mathbb{U}$, i.e., mapping each row of $\mathbf{M}$ to an attribute in $\mathbb{U}$.
- Assume $\mathbb{A}$ is a set of attributes and $I = \{i | i \in \{1, ..., n_1\}, f(i) \in \mathbb{A}\}$ be the set of rows in $\mathbf{M}$ that belong to $\mathbb{A}$. We say that $(\mathbf{M}, f)$ accepts $\mathbb{A}$ if there exists a linear combination of rows in I that gives (1,0,...,0). This can be done by asking Alice to input some attributes which are transformed into $(\mathbf{M}, f)$ at the Eunomia app on Alice's smartphone.

The data provider encrypts data as follows:

- dp_i picks $u_1, u_2 \leftarrow_R \mathbb{Z}_p$ and calculate $ct_i^0 := (A_1^{u_1}, A_2^{u_2}, g_2^{u_1+u_2})$; for $j = 1, ..., n_1$ and $v = 1$, 2, 3, calculate $ct_{i,v} = H_1(f(j)v1)^{u_1} \cdot H_1(f(j)v2)^{u_2} \cdot \prod_{o=1}^{n_2} [H_1(0ov1)^{u_1} \cdot H_1(0ov2)^{u_2}]^{(\mathbf{M})_{j,o}}$.
- dp_i sets $ct_i := (ct_{i,1}, ct_{i,2}, ct_{i,3})$, calculates $ct_i' = B_1^{u_1} \cdot B_2^{u_2} \cdot d_i$, sets the encrypted data $Ed_i = (ct_i^0, ct_i^1, ..., ct_i^{n_1}, ct_i')$. Here, CP-ABE guarantees that only qualified data requesters with corresponding attributes can access the data later.

The data provider prepares an anonymous credential as follows:

- dp_i accesses the cb to obtain a set of valid anonymous credentials $\mathbb{C}_i = \{cr_1, cr_2, ..., cr_{n_3}\}$ including dp_i's credential cr_j, calculates $A_i = s^{cr_1 cr_2 \cdots cr_{n_3}} \bmod N$ and $w_i = s^{cr_1 cr_2 \cdots cr_{n_3}/cr_i} \bmod N$.
- dp_i computes a proof of knowledge: $\pi_i' = \text{NIZKPoK}\{(sk_i, w_i, r_i', cr_i, r_i, pi_i) : \mathsf{Verify}(pp., A_i, cr_i, w_i) = 1 \wedge cr_i = h_0^{r_i'} h_1^{sk_i} \prod_{o=0}^{m} h_{o+2}^{att_i^o} \wedge pi_i = h_0^{r_i} h_1^{sk_i}\}$ where the $\mathsf{Verify}(pp., A_i, cr_i, w_i)$ computes $A' = w_i^{cr_i}$ and outputs 1 if and only if $A' = A_i$. Therefore, md_i is ("out-vehicle", "photo", 20200523, $21:30$). The zero-knowledge proof is used to prove the validity of the anonymous data uploading.

The data provider uploads data as follows:

- dp_i computes two hash values $HV_i^1 = H_2(pi_i || \pi_i || \mathbb{C}_i || md_i)$ and $HV_i^2 = H_2(Ed_i)$.
- dp_i sends Ed_i to the off-chain DDSS and submits md_i to the on-chain cb by sending a data uploading transaction signed with private key prk_i to complete data uploading

$$Tx_3 = [''\text{Upload}'', pi_i, \pi_i', \mathbb{C}_i, md_i, HV_i^1, HV_i^2]_{prk_i}, \tag{3}$$

where prk_i is obtained from the Ethereum platform and it has to be changed in each data uploading to prevent linkability.
- When a blockchain node receives Tx_3, it checks its validity by computing $A' = w_i^{cr_i}$ and verifying that π_i is the proof of knowledge on cr_i, $\mathbb{C}_i$ and pi_i.

5.5 Data Accessing

A crime police investigator accesses data as follows:

- A crime police investigator cpi_j leads the investigation on a case X wishes to obtain some data d_i from the cb. To the end, cpi_j first builds a metadata md_j according to the investigation requirements. For example, cpi is looking for some witness who has a video clip of a vehicle crashing into a crowd around 8 A.M. on May 26, 2020. Then cpi_j can set md_j as ("out-vehicle", "short video", 20200526, $08:00$).
- cpi_j constructs a range proof: $\pi_j'' = \{(wa_j, \tilde{r}) : \tilde{R} = g_1^{\tilde{r}} g_2^{wa_j} \wedge wa_j \in [0, 32767]\}$ ($32767 = 2^{15} - 1$, the smallest power of 2 bigger than 21526). This can be instantiated via a noninteractive bulletproof system using the Fiat-Shamir heuristic [31]. The zero-knowledge proof is used to prove the validity of the warrant.
- cpi_j prepares a similar $(pi_j, \pi_j', \mathbb{C}_j, md_j)$ for anonymous authentication. cpi_j compute a hash value $HV_j = H_2(pi_j || \pi_j' || \mathbb{C}_j || md_j || \pi_j'')$ and sends a data access request to cb by sending a data accessing transaction

$$Tx_4 = [''\text{Access}'', X, pi_j, \pi_j', \mathbb{C}_j, md_j, \pi_j'', HV_j]_{prk_j}. \tag{4}$$

5.6 Data Retrieval

The data requester sends a data retrieval transaction and to get the encrypted data as follows:

- After receiving Tx_j from cpi, a blockchain node au verifies the validity of π_j and π_j'. If the two proofs are successfully verified, au returns a download link L and a signature σ to cpi. Then au sends a permission transaction to cb to allow cpi_j to retrieve data from CPD

$$Tx_5 = [''\text{Permit}'', X, au, \pi_j, date, ts]_{prk_{au}}. \tag{5}$$

- cpi_j signs (L, σ) with her/his master private key $\overline{prk}_j$ and sends $(cpi_j, \overline{puk}_j, (L, \sigma)_{\overline{prk}_j})$ to the CPD which retrieves the corresponding data Ed from the DDSS. Then cpi_j and the CPD execute a committed user oblivious transfer protocol where cpi_j is the receiver and CPD is the sender. The oblivious transfer ensures that the crime police investigator embeds his key in the requested data while not showing the key to the crime police department.
- CPD decrypts Ed using a corresponding attribute key to obtain data d. CPD breaks d into τ data blocks db_i, $0 \le i \le \tau - 1$ for a bit τ-long private key $\overline{prk}_j$. Each data block is watermarked to produce two versions db_i^0, db_i^1.
- CPD chooses a random $r \leftarrow_R \mathbb{Z}_p$ and sends g_1^r to cpi_j.
- For all $0 \le i \le \tau - 1$, cpi_j chooses a random $r_i \leftarrow_R \mathbb{Z}_p$, computes a commitment $\mathsf{C}_i = g_1^{r_i} g_1^{rb_i}$, $\mathsf{R} = \sum_{i=0}^{\tau-1} 2^i r_i$ where $\hat{b_i}$ is the bit decomposition of $\overline{prk}_j$, and sends $(g_1^{\mathsf{R}}, \mathsf{C}_i, \mathsf{R}, \text{NIZK}\{(r_i, \hat{b}_i) | g_1^{r_i} g_1^{rb_i}\})$ to CPD. By doing so, cpi_j has proved the possession of $\overline{prk}_j$. CPD computes $\mathsf{C} = \Pi_{i=0}^{\tau-1} \mathsf{C}_i^{2^i}$. CPD aborts if $\mathsf{C} \neq (g_1^{\mathsf{R}} \overline{puk}_j^r)$ or the NIZK fails.
- For all $0 \le i \le \tau - 1$ and $j = 0, 1$, CPD computes $k_{i,j} = H_3((\mathsf{C}_i \cdot (g_1^r)^{-j})^r)$ and sends $\mathsf{P}_i = H_3(H_2(k_{i,0})) \oplus H_3(H_2(k_{i,1}))$ to cpi_j. cpi_j computes $k_{i,\hat{b}_i} = H_3((g_1^r)^{r_i})$ and sends $\mathsf{P}_i' = H_3(H_2(k_{i,\hat{b}_i})) \oplus \mathsf{P}_i \hat{b}_i$ to CPD.

– CPD verifies whether $\mathsf{P}'_i = H_3(H_2(k_{i,0}))$, computes $Ed'_i = \mathsf{Encrypt}(pk, \mathbb{S}_{dp}, db_i^j)$, and sends ciphertext $C_i = \mathsf{Enc}(Ed'_i)$ to cpi_j where Enc is symmetric encryption. CPD sends a data retrieval transaction to cb to allow cpi_i to start analyzing the data

$$Tx_6 = [''\text{Analyze}'', X, \text{CPD}, \pi_j, date, ts]_{prk_{CPD}}. \quad (6)$$

After receiving $\{C_i\}$, $0 \le i \le \tau - 1$, cpi_j first decrypts C_i using $k_{i,\hat{b}_i}$ and Dec. Then cpi_j rebuilds Ed' with $\{Ed'_i\}$. Next, cpi_j decrypts Ed' using attribute key sk_j^A as follows.

– If the set of attributes $\mathbb{A}_j$ in sk_j^A satisfies the MSP $(\mathbf{M}, f)$ in Ed', then there exists constants $\{\lambda_i\} \in$ that satisfies $\prod_{i\in I} \lambda_i(\mathbf{M})_i = (1, 0, ..., 0)$.
– cpi_j computes $d_j^1 = ct' \cdot e(\prod_{i\in I}(ct_1^i)^{\lambda_i}, sk_j^{11}) \cdot e(\prod_{i\in I}(ct_2^i)^{\lambda_i}, sk_j^{12}) \cdot e(\prod_{i\in I}(ct_3^i)^{\lambda_i}, sk_j^{13})$ and $d_j^2 = e(sk_j^{1'} \cdot \prod_{i\in I}(sk_j^{f(i),1})^{\lambda_i}, ct_1^i) \cdot e(sk_j^{2'} \cdot \prod_{i\in I}(sk_j^{f(i),2})^{\lambda_i}, ct_2^i) \cdot e(sk_j^{3'} \cdot \prod_{i\in I}(sk_j^{f(i),3})^{\lambda_i}, ct_3^i)$.
– cpi_j obtains the requested data by computing $d = d_j^1/d_j^2$.

Note that, in order to protect the CPD from knowing the d, we can use a secure enclave, such as Intel Software Guard eXtensions (SGX) [38], to realize the "decrypt-watermark-encrypt" process conducted by the CPD. It is a potential research direction that is a part of our work. When the data type is taken into account, the decrypt-watermark-encrypt can be realized by first transforming the data into a photo and processing it with the SGX module. The process of data uploading, data accessing, and data retrieval is shown in Fig. 3.

5.7 Data Analysis and Result Reporting

After all the related data is collected, cpi_j conducts systematic analysis on the data using professional forensic techniques, softwares and hardware. When a report $\mathcal{R}_j$ is finalized, cpi_j sends a report transaction to cb to complete the analysis

$$Tx_7 = [''\text{Report}'', X, H_2(\mathcal{R}_j), date, ts]_{prk_j}. \quad (7)$$

The analysis aims to locate data that can be presented as potential pieces of evidence, such as a photo of a suspect being at the scene, braking distance, and a surveillance video of withdrawing money. These evidences are used to substantiate a criminal activity and support the recommendation of prosecution. Besides, cpi_j can send the data to a legal medical expert for data analysis. In this case, the Tx_7 has to include the anonymous credential information of the expert and cpi_j has to undertake the due obligations if the expert leaks data. To mitigate this effect, we assume the investigator has the expertise to conduct data analysis in this work.

Afterward, cpi_j proceeds with the preparation of a digital report detailing the findings of the analysis. Finally, cpi_j sends a close transaction to cb to put an investigation into closure

$$Tx_8 = [''\text{Close}'', X, date, ts]_{prk_j}. \quad (8)$$

5.8 Traitor Tracing

Upon revelation of some watermarked data db, the data user's private key prk is extracted from the data and shown to the authority with which she/he retrieves data. The authority will expose the real identity of the data user and punish the user according to legal provisions.

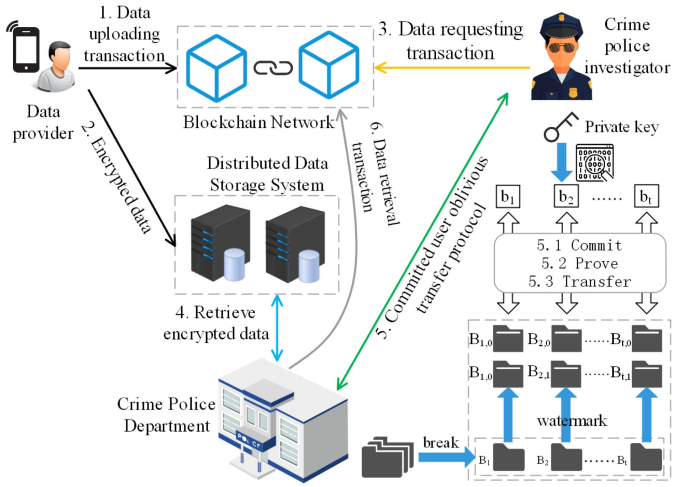


Fig. 3. Overview of data uploading, data accessing, and data retrieval.

6 Privacy and Security Analysis

In this section, we formally prove that Eunomia achieves all the privacy and security goals defined in Section 3C.

6.1 Privacy

6.1.1 Anonymity

Theorem 1. *Eunomia guarantees anonymity under the Discrete Logarithm (DL) assumption.*

Proof. The idea behind the proof of this theorem is that if it were possible to recover the master secret from an anonymous credential cr, then it would also be possible to break the DL assumption [39]. We formalize it through a proof by security reduction, where we show how to use any efficient adversary $\mathcal{A}$ to construct another adversary $\mathcal{A}'$ such that if $\mathcal{A}$ violates the theorem, then $\mathcal{A}'$ violates the DL assumption. Since the DL problem is assumed to be hard, it must also be secure in the sense of the theorem. Construct the following adversary $\mathcal{A}'$:

1. Upon observing $\mathbb{G}, q, h_1$, generate $h_0, h_2, .., h_n$, r, r', sk, att, aux satisfying $\mathbb{G} = \langle h_0 \rangle = \langle h_2 \rangle = ... = \langle h_n \rangle$.
2. Compute $cr = h_0^{r'} h_1^{sk} \prod_{o=0}^{m} h_{o+2}^{att_i^o}$, compute $\pi = \text{ZKSoK}[aux]\ \{(sk, r', r) : cr = h_0^{r'} h_1^{sk_i} \prod_{o=0}^{m} h_{o+2}^{att^o} \wedge pi = h_0^r h_1^{sk}\}$.
3. Invoke $\mathcal{A}(k, \mathbb{G}, q, h_0, h_1, ..., h_n, cr, \pi, pi)$.
4. Output $sk'_{\mathcal{A}}$ after $\mathcal{A}$ outputs $sk'_{\mathcal{A}}$.

$\mathcal{A}'$ runs in polynomial time since $\mathcal{A}$ does. By the definition of the discrete-logarithm experiment $\mathsf{DLog}_{\mathcal{A}',\mathcal{G}}(k)$ [36], we can see that $\mathcal{A}'$ succeeds if and only if $\mathcal{A}$ outputs a correct $sk'_{\mathcal{A}}$. So $\Pr[\mathsf{DLog}_{\mathcal{A}',\mathcal{G}}(k) = 1] = \Pr[\mathcal{A}(k, \mathbb{G}, q, h_0, h_1, ..., h_n, cr) = sk]$. Given that the DL problem is hard relative to $\mathcal{G}$ for all PPT adversaries, there is a negligible function negl such that $\Pr[\mathsf{DLog}_{\mathcal{A}',\mathcal{G}}(k) = 1] \le \text{negl}(k)$. We conclude that $\Pr[\mathcal{A}(k, \mathbb{G}, q, h_0, h_1, ..., h_n, cr) = sk] \le \text{negl}(k)$. Due to the DL assumption, it is hard to compute sk from pi. Due to the zero-knowledge property of Bulletproof [21] relying only on the DL assumption, the warrant is not shown to

the verifier either, i.e., $|\Pr[\mathsf{Exp}^{\mathrm{ano}_2}_{\mathcal{A},\Pi}(k)=1| \leq \mathrm{negl}(k)$. We complete the proof. □

6.1.2 Unlinkability

Theorem 2. *Eunomia guarantees unlinkability under the DL assumption, Decisional Linear (DLIN) assumption and randomness.*

Proof. For the anonymous credential cr, pseudo-identity pi, and zero-knowledge proof π', all of them are generated under the DL assumption and with fresh random numbers. For the ciphertext Ed, it is computed under the DLIN assumption with fresh random numbers u_1, u_2. Therefore, from the view of $\mathcal{A}$, they are drawn from a uniform distribution such that $\mathcal{A}$ can only distinguish the two messages with a probability that is not any better than taking a random guess, i,e., $|\Pr[\mathsf{Exp}^{\mathrm{unl}_1}_{\mathcal{A},\Pi}(k)]=1| \leq \frac{1}{2}+\mathsf{negl}(k)$ and $|\Pr[\mathsf{Exp}^{\mathrm{unl}_2}_{\mathcal{A},\Pi}(k)]=1| \leq \frac{1}{2}+\mathsf{negl}(k)$. We complete the proof.□

6.2 Security

6.2.1 Confidentiality

Theorem 3. *Eunomia protects data confidentiality under the DLIN assumption on asymmetric pairing groups in the random oracle model.*

Proof. The proof is constructed through a sequence of hybrids and a hybrid describes how $\mathcal{C}$ interacts with $\mathcal{A}$. The first hybrid HB_1 is $\mathsf{Exp}^{\mathrm{con}}_{\mathcal{A},\Pi}(k)$ in the Appendix with one difference that H_1 is assumed to be a random oracle. The second hybrid HB_2 is a modified version of HB_1. In initialization, $p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2$ are generated the same. $\mathcal{C}$ maintains two lists L_1 and L_2 to simulate the random oracle. L_1 is in the form $(x, \mathbf{W}_x)$ or $(y, \mathbf{U}_y)$. x is an arbitrary binary string, y is a positive integer, and $\mathbf{W}_x$, $\mathbf{U}_y$ are 3×3 matrices. L_2 has entries of the form (a, b) where a is either xlt or $0ylt$ (for $l \in \{1,2,3\}$ and $t \in \{1,2\}$) or others, and $b \in \mathbb{G}_1$. $\mathcal{A}$ makes three types of oracle queries: xlt, $0ylt$, and others. In key generation, $\mathcal{A}$ sends a key query KQ to $\mathcal{C}$, $\mathcal{C}$ retrieves $\mathbf{W}_x$ for every $x \in KQ$ and $\mathbf{U}_1$ from L_1 and returns an attribute key to $\mathcal{A}$. In encryption, $\mathcal{A}$ sends two messages m_0, m_1, and $(\mathbf{M}, f)$ to $\mathcal{C}$, $\mathcal{C}$ retrieves $[(\mathbf{W}^T_{f(i)}\mathbf{A})_{l,t}]_1$ and $[(\mathbf{U}^T_j\mathbf{A})_{l,t}]_1$ for all $i = 1, ..., n_1, j = 1, ..., n_2, l, t$ from L_2. $\mathbf{A}$ is a 3×2 matrix $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)$ and $\mathbf{a}_1 = (r_1, 0)$, $\mathbf{a}_2 = (0, r_2)$, and $\mathbf{a}_3 = (1, 1)$. $\mathcal{C}$ computes a ciphertext and returns it to $\mathcal{A}$. □

From the view of $\mathcal{A}$, HB_1 and HB_2 are statistically indistinguishable because $\mathbf{W}_x$, $\mathbf{U}_y$ have enough entropy to make $(\mathbf{W}^T_{f(i)}\mathbf{A})_{l,t}$ and $(\mathbf{U}^T_j\mathbf{A})_{l,t}$ look random for each l, t. Then, a set of hybrids can be defined with the last HB_6 putting the randomness to the ciphertext such that the encryption of any message is indistinguishable from the encryption of a random message under the DLIN assumption. It can be proven that each hybrid is indistinguishable from the next one. In the end, HB_1 is indistinguishable from HB_6. For more details, please refer to [20].

6.2.2 Authentication

Theorem 4. *Eunomia protects authentication under the DL and Strong RSA assumptions.*

Proof. First, we begin by discussing the signature/proof π. Let us assume that $\mathcal{A}$ has non-negligible advantage ϵ in winning the experiment $\mathsf{Exp}^{\mathrm{aut}}_{\mathcal{A},\Pi}(k)$. Then we can construct an adversary $\mathcal{A}'$, which has the same advantage ϵ in breaking the DL problem. Construct the following $\mathcal{A}'$:

1. Upon observing $\mathbb{G}, q, h$, satisfying $h = g^x$ for some x, generate $h_0, h_2, .., h_n, r, r', sk, att, aux$.
2. Compute $cr = h_0^{r'} h_1^{sk} \prod_{o=0}^{m} h_{o+2}^{att^o_i}$, compute $\pi = \mathrm{ZKSoK}[aux]\ \{(sk, r', r) : cr = h_0^{r'} h_1^{sk_i} \prod_{o=0}^{m} h_{o+2}^{att^o} \wedge pi = h_0^r\, h_1^{sk}\}$.
3. Invoke $\mathcal{A}(k, \mathbb{G}, q, h_0, h_1, ..., h_n, cr, \pi, pi)$.
4. $\mathcal{A}$ outputs $(sk_{\mathcal{A}}, r_{\mathcal{A}}, r'_{\mathcal{A}})$, output $(sk_{\mathcal{A}}, r_{\mathcal{A}}, r'_{\mathcal{A}})$.

When $\mathcal{A}$ outputs $(sk_{\mathcal{A}}, r_{\mathcal{A}}, r'_{\mathcal{A}})$ that is different from (sk, r, r'), we say that $\mathcal{A}$ successfully forges a valid π, i.e., breaks the binding property. Since $\mathcal{A}'$ runs in polynomial time, $\mathcal{A}'$ identifies a collision on the commitments which contradicts with the DL assumption. Therefore, ϵ should be negligible. □

Second, we discuss the signature/proof π' by giving an intuition behind the proof. π' in data uploading/accessing is a statistical non-interactive zero-knowledge proof of knowledge of (sk, w, r', cr, r, pi) such that w is a witness that cr is in the accumulator A and both pi and cr belong to sk. π is generated via leveraging standard techniques as well. To forge a valid π', $\mathcal{A}$ has to find a collision on the commitments or forge an accumulator membership proof, which occurs with negligible probability under the Strong RSA and Discrete Logarithm assumptions. We complete the proof.

6.2.3 Fine-Grained Access Control

If a data user that does not possess enough attributes satisfying the access structure $\mathcal{S}$ tries to access d, the set of attributes A in her/his sk do not satisfy the MSP (M, f), thus failing to decrypt the ciphertext of d. For the warrant, we require that each warrant has a period of validity. After the warrant expires, a malicious CPI cannot use the invalid warrant to continue investigation. Hence, fine-grained access control is achieved.

6.2.4 Accountability

As shown in Fig. 2, we model the process of a VDF investigation as a finite state machine in which each state transition is realized via a blockchain transaction, including a signature by the transaction initiator. The signature has ensured the integrity of data. Specifically, adversaries cannot interfere with the execution of smart contracts and prompt transition without being permitted or verified. Meanwhile, the encrypted data is stored in the DDSS while its hash is recorded on the blockchain. The DDSS and the data owner can audit the data by recomputing a hash value of local data and comparing it with the on-chain hash value. Hence, the system can protect the digital forensics chain of custody so that the data is hard to be tampered with and easy to be audited, providing accountability.

6.2.5 Traceability

During data retrieval with an authority, the data user has to embed the private key $\overline{prk}$ into the request data d via a committed user oblivious transfer protocol before receiving the

TABLE 2
Comparison of Privacy and Security Properties

Property	Anonymity	Unlinkability	Confidentiality	Authentication	Access Control	Accountability	Traceability
DiaLOG [25]			✓	✓			
DFAV [26]			✓			Partial	
B4F [6]	✓		✓	✓		✓	
BB-VDF [12]			✓	✓	✓	✓	
Eunomia	✓	✓	✓	✓	✓	✓	✓

data. Since the zero-knowledge proof is sound, the data user cannot deceive the authority to forge a valid private key and obtain the requested data. Specifically, the authority requires the data user to make a commitment of each choice $\hat{b_i}$ as $\mathsf{C}_i = g_1^{r_i} g_1^{r\hat{b}_i}$ and proves that the bits $\{\hat{bi}\}_{i=0}^{\tau-1}$ can be used for recovering the $\overline{prk}$. Hence, the system can trace the leak source by extracting the private key according to the leaked data blocks, achieving traceability.

We compare Eunomia with existing work regarding privacy and security in Table 2. Only Eunomia provides all the designed objectives. Specifically, DiaLOG [25] allows data confidentiality by using symmetric encryption in registration and requesting. It guarantees integrity by using signatures in data retrieval. DFAV [26] encrypts data by using symmetric encryption. Then it encrypts the key by using asymmetric encryption and the public key of the database. Partial accountability is provided due to the hash value submitted by the data provider while it is not publicly verifiable. B4F [6] uses the IEEE 1609.2 standard to ensure security. However, they mention that disclosure of user data is necessary due to regulations and policies. BB-VDF [12] has adopted a similar forensics framework to address confidentiality, authentication, access control, and accountability. All the work above cannot track the source of data leakage under a data breach.

7 Performance Evaluation

In this section, we build a prototype in a WiFi-based Ethereum test network to evaluate its computational costs, communication overhead, network delay, monetary cost, power consumption, and scalability. Given the experimental results from these aspects, we see that the whole system is executable and practical.

7.1 Experiment Settings

The blockchain test platform is the Ethereum[2], and we use Geth (Go-Ethereum Client)[3] as the main tool for Ethereum network environment establishing. We instantiate four authorities on four nodes, each node has a signer account with authority to seal a block. One node with a public IP address is the boot node. The other three nodes are deployed on two laptop computers and one desktop computer. We instantiate data provider/user on two smartphones. We use JPBC library[4] to implement cryptographic primitives with an elliptic curve being defined as $y^2 = x^3 + x$ over $\mathbb{F}_q$. $|p| = 512$ is a value from the type A pairing in JPBC[5]. $|q| = 1024$ is decided to achieve a strong security level. We choose SHA256 as the hash function and AES for symmetric encryption for their recognition and wide adoption in cryptography protocols. H_1 uses SHA256 to convert the input to a hash value and maps the value to an element in $\mathbb{G}_1$. The m is set to 10 to the include main attributes of an investigation, which is determined after consulting with the local police department in Hefei, China. They include type, date, time, address, etc. We require that a cpi must possess all the attributes to decrypt encrypted evidence, i.e., $|\mathbb{A}| = 10$. The value of n_1 and n_2, as the size of a matrix, are decided by m. $n_3 = 10$ is desinged to offer data providers several chances to upload evidence. $aux = 10$ is a value from the ZKSoK. $|\tau| = 1024$ is the size of the cpi_j's $\overline{prk}_j$.

We use the Puppeth (Ethereum Private Network Manager[6]) imported from Geth[7] to create the genesis block. The consensus mechanism is Clique (Proof-of-Authority) and the block creation time is set as 10 seconds. Since each transaction costs different amount of gas and the gasLimit is fixed (4700000 per block), the number of transactions in a block varies. After activation, each blockchain node will generate enode information (uniform resource identifier), replace "0.0.0.0" with its virtual IP, and then invoke *admin.addPeer()* to record other nodes' enode information.

We write the smart contract with an online integrated development environment remix[8] and deploy it with a Chrome Ethereum wallet plugin metamask[9] via injected web3 protocol. After deployment, we record the generated contract address and application binary interface, then invoke the contract via web3 protocol. For each investigation, there is an enumeration struct named *State* to store the state. Once the transaction function is triggered, the Ethereum smart contract first checks whether the pre-state of investigation is legal by using Solidity statements "require". If the pre-state is legal, the state is transited to the next state after all operations in the transaction are completed correctly.

Figs. 4 and 5 show the overview of Eunomia framework and detailed information flow of Eunomia. Key experimental parameters are listed in Table 3. Source codes[10] are uploaded to GitHub. We also conduct experiments on two smartphone as data user and data provider. We develop an Android application, use the Springboot framework for the server, choose Java to write action logic with JPBC library, and enable

2. https://www.ethereum.org
3. https://github.com/ethereum/go-ethereum
4. http://gas.dia.unisa.it/projects/jpbc
5. http://gas.dia.unisa.it/projects/jpbc/docs/ecpg.html#.YWJebmJBzZR
6. https://github.com/ethereum/go-ethereum/tree/master/cmd/puppeth
7. https://github.com/ethereum/go-ethereum
8. https://remix.ethereum.org
9. https://metamask.io
10. https://github.com/SopmmmodII/Lawful-Evidence-Management

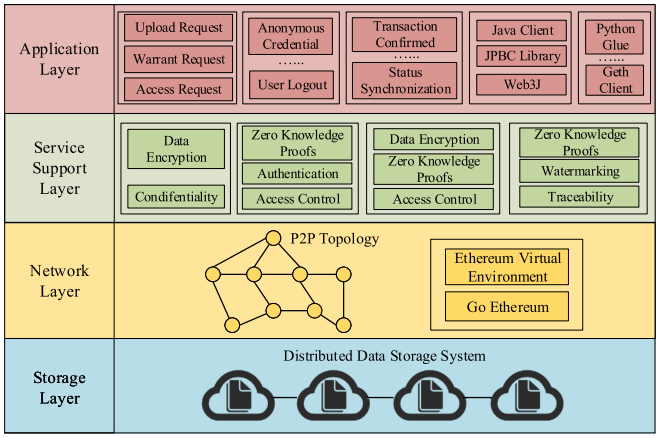


Fig. 4. Overview of Eunomia framework.

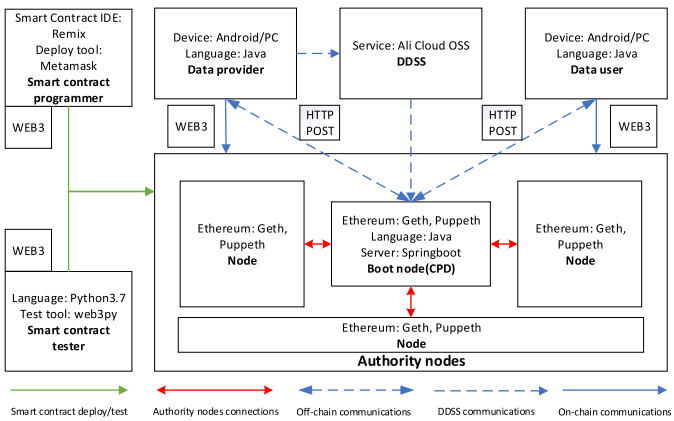


Fig. 5. Detailed information flow of Eunomia modules.

communications between the smartphone and the blockchain via http post protocol. AliCloud Object Storage Service OSS is chosen by us as DDSS. We choose Python 3.7 and web3py library to perform our test operations. Three screenshots of the implementation of Eunomia are shown in Fig. 6.

7.2 Computational Costs

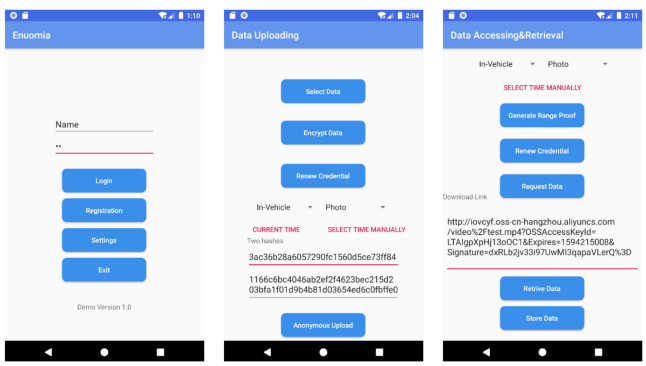
We now analyze the computational costs for the four main entities: DP, DU, AU, and BN through counting the total number of cryptographic operations. Different notations are adopted to denote cryptographic operations in the scheme. G_1 and G_2 denote the multiplication and exponentiation in $\mathbb{G}$. G_1' and G_2' denote the multiplication and exponentiation in $\mathbb{G}_1$ ($\mathbb{G}_2$). H_i refers to the specific hash function, i.e., H_1, H_2, and H_3. bp denotes the bilinear pairing in CP-ABE. G_{T1}, G_{T2}, and G_{T3} denote the multiplication, exponentiation, and division in $\mathbb{G}_T$. wm refers to the watermarking.

A data provider's primary computational cost rests in entity registration and data uploading. It consists of generating a signature of knowledge π, encrypting data d, computing a proof of knowledge π_i', and computing two hashes HV^1 and HV^2, i.e., $((m+6)G_1+(m+7)G_2+H_2)+(3G_2'+n_1(2+2n_2)H_1+2G_{T1}+2G_{T2})+(3G_1+4G_2+H_2)+2H_2$. In entity registration and data uploading, it costs the data provider 120 ms and 44 ms, respectively.

A data user's primary computational cost lies in entity registration, data accessing, and data retrieval. It consists of

TABLE 3
Key Experimental Parameters

Experimental Parameters	Value
$\|p\|$, $\|q\|$; H_1^*, H_2, H_3	512, 1024; SHA256
m, n_1, n_2, n_3	10, 10, 10, 10
$\|\mathbb{A}\|$, aux, $\|\tau\|$	10, 10, 1024
Enc/Dec	AES Encryption/Decryption
*Map the output of SHA256 to $\mathbb{G}_1$.	
Desktop computer	Intel i3 7100, 4-Core 8 GB/Windows 10 Professional
Laptop computer	Intel i5 10210u, 4-Core 8 GB/Windows 10 Professional
Ali cloud server [22]	Intel Xeon E5 2682 v4, 2 Core of 16 4 GB/Ubuntu16.04
Smartphones	Huawei Kylin 980 8 GB/Android 9 "Pie"



(a) Login (b) Data Uploading (c) Access&Retrieval

Fig. 6. Implementation of Eunomia using smartphones.

generating a signature of knowledge π, constructing a range proof π'', generating an anonymous credential $(pi_j, \pi_j', \mathbb{C}_j, md_j)$, computing a hash value HV, signing (L, σ), generating a proof, decrypting C and Ed', and computing one hash of the report, i.e., $((m+6)G_1+(m+7)G_2+H_2)+(2G_1'+5G_2'+H_2)+(3G_1+4G_2+H_2)+H_2+(2\tau G_2'+\tau(G_1+2G_2'+H_2)+\tau H_2+\tau H_3)+(\tau\mathsf{Dec}+6bp+4G_{T1}+G_{T3})+H_2$. In entity registration, data accessing, and data retrieval, it costs the data user 120 ms, 109 ms, and 9668 ms, respectively. Although the data retrieval has a relatively high computational cost raised from multiple zero-knowledge proofs, it is still worth the time if we aim to guarantee traitor tracing.

The authority in entity registration processes the registration request from a data user, namely computing an attribute key sk^A, i.e., $2G_1'+(2*|\mathbb{A}|+6)G_2'+(6*|\mathbb{A}|+2)H_1$. The authority in data retrieval participates in the committed user oblivious transfer protocol, i.e., $(6bp+4G_{T1}+G_{T3})+(2\tau wm+G_2'+(\tau-1)G_1'+\tau G_2'+2\tau G_1'+3\tau G_2'+3\tau H_2+3\tau H_3)+(3G_2'+n_1(2+2n_2)H_1+2G_{T1}+2G_{T2}+\tau\mathsf{Enc})$. The watermark embed/extract operation using Fast Fourier Transform (FFT) is executed $2|\tau|$ times for one picture.

A blockchain node verifies the validity of entity registration transaction, warrant grant transaction, data uploading transaction, data accessing transaction, permission transaction, data retrieval transaction, data analysis transaction, and result report transaction. For the first four transactions, the node has

TABLE 4
Implemented Running Time (Unit: Millisecond)

Phase	Entity Registration				Data Uploading		Data Accessing		Data Retrieval			Result Reporting		Traitor Tracing
Entity	DP	DU	AU	BN	DP	BN	DU	BN	AU	DU	BN	DU	BN	CPD
DiaLOG[1] [25]	n/a	0.23	0.23	n/a	n/a		n/a		15[2]	0.23	n/a	n/a		n/a
DFAV [26]	n/a				16	n/a	n/a		0.23[3]	0.23	n/a	n/a		n/a
B4F [6][4]	n/a	n/a	< 1[5]	n/a	< 1	< 1	n/a		n/a			n/a		n/a
BB-VDF [12][6]	n/a				43	15	20	80[7]	85	230	20	20	20	n/a
Eunomia	120	120	193	94	44	94	109	86	5066	9668	4	< 1	4	1400/500[8]

[1] *We choose an AES key of 128-bit, an RSA key of 1024-bit, and a message of 160-bit in the scheme DiaLOG.* [2] *The AU in the scheme DiaLOG is a central communication unit.* [3] *The AU in the scheme DFAV is a database.* [4] *We use Elliptic Curve Integrate Encrypt Scheme (ECIES) for encryption and signing in the scheme DFAV.* [5] *The AU in the scheme B4F is a registration authority.* [6] *We use AES-128 for the symmetric encryption in the scheme BB-VDF.* [7] *In the scheme BB-VDF, the data access transaction is sent to a court.* [8] *1400 is for embedding a watermark, 500 ms is for extracting a watermark. For example, a watermark is a random ID: "93498ac8978f".*

to not only verify the signature, but also verify the proofs included in the transaction. Specifically, besides verifying the signature, it verifies ac in Tx_1 and Tx_2 (i.e., $2G_1 + 6G_2 + 2H_2$), verifies π' in Tx_3 (i.e., $2G_1 + 6G_2 + 2H_2$), and verifies π' and π'' in Tx_4 (i.e., $G_1 + 2G_2 + 2G'_1 + 4G'_2 + 2H_2$). The verification of a signature costs 4 ms. The authorities and blockchain nodes are assumed to possess powerful computation powers, so the computational costs can be further reduced.

The main measured values for execution time of each entity are listed in Table 4. The computational costs in the first three phases are relatively low, i.e., less than 200 ms for each entity in each phase, for using lightweight anonymous authentication and lightweight CP-ABE. While the DU and AU spend more time finishing the transfer of evidence because they have to conduct oblivious transfer, prove and verify the possession of a private key, and decrypting and encrypting the data. Although Eunomia has extra computation burden, still the recorded runtime of each entity fall in an acceptable range.

7.3 Communication Overhead

A data provider sends an entity registration transaction Tx_1 to the blockchain network, i.e., $|''\text{Register}''| + |au| + |ac_i| + |date| + |ts| + |signature| = |''\ Register''| + |au| + |\mathbb{G}| + 2 * (|q| * 2 + |p| * 2) + |att_i| + |pi_i| + |aux_i| + |date| + |ts| + |signature| = 8 * 8 + 4 + 1024 + 2 * (1024 * 2 + 512 * 2) + 2 * 10 + 10 + 10 + 20 + 10 + 520 = 4750$ bits = 0.58 Kbytes. The data provider also sends an data uploading transaction Tx_3, i.e., $|''\text{Upload}''| + |pi_i| + |\pi'_i| + |\mathbb{C}_i| + |md_i| + |HV_i^1| + |HV_i^2| + |signature| = 6 * 8 + 10 + (1024 + 1024 + 1024 + 1024 + 2 * (2048 + 256)) + 10 * 1024 + (10 + 2 + 20 + 10) + 2 * 256 = 20076$ bits = 2.45 KBytes. A data user sends an anonymous credential to the AU, submits a data accessing transaction Tx_4, a report transaction Tx_7, and a close transaction Tx_8, i.e., $|cr_j| + |\pi_j| + |att_j| + |pi_j| + |aux_j|$ = 0.88 KBytes, $|''\text{Access}''| + |pi_j| + |\pi'_j| + |\mathbb{C}_j| + |md_j| + |\pi''_j| + |HV_j| + |signature|$ = 2.95 KBytes, $|''\text{Report}''| + |H_2(\mathcal{R}_j)| + |date| + |ts| + |signature|$ = 0.10 KBytes, and $|''\text{Close}''| + |X| + |date| + |ts| + |signature|$ = 0.08 KBytes. The data user transfers 0.56 KBytes in retrieving data from the AU. The authority in entity registration returns an attribute key $(sk_j^1, \{sk_j^x\}_{x \in \mathbb{A}_j}, sk_j^3)$ to DU, i.e., $|sk_j^{11}| + |sk_j^{12}| + |sk_j^{13}| + |sk_j^{x,1}| + |sk_j^{x,2}| + |sk_j^{x,3}| + |sk_j^{1'}| + |sk_j^{2'}| + |sk_j^{3'}| = 1.125$ KBytes, and submits a warrant grant transaction Tx_2 similar to Tx_1, i.e., 0.58 KBytes. The authority submits a permission transaction Tx_5, i.e., 0.64 KBytes, sends 16.15 KBytes in the committed user oblivious transfer protocol, and a data retrieval transaction, i.e., 0.64 KBytes. The original picture we use is 36.1 KB per data block and the embedded one is 73.8 KB. A blockchain node verifies each transaction that is submitted to the blockchain network. The length of the signature in each transaction is 520-bit. Eunomia's communication overhead is higher in data uploading, data retrieval, and traitor tracing. They are also the foreseeable cost of enhancing privacy and security.

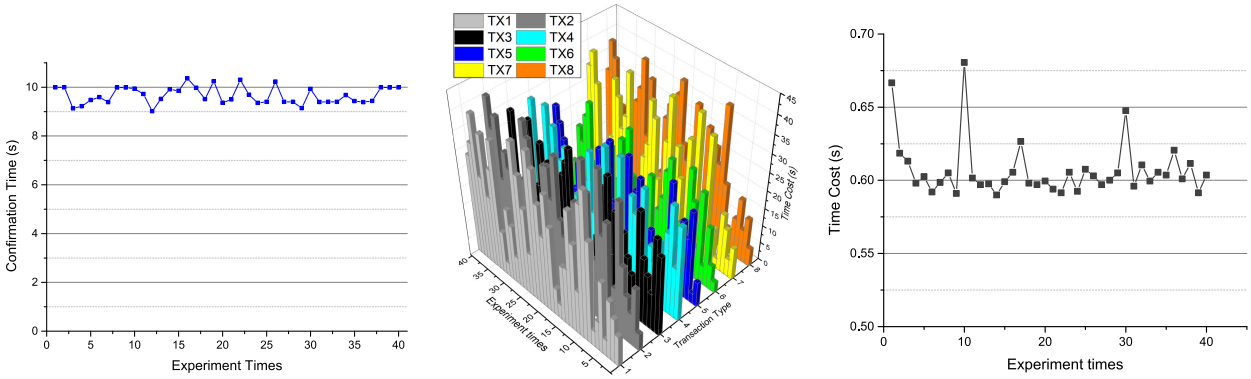
7.4 Network Delay

We set the block creation time, i.e., block period, to be 10 seconds. Experimental results in Fig. 7a indicate that the real consensus time fluctuates around 10 seconds due to the hardware interference. Next, we test the transaction confirmation time by using *waitForTransactionReceipt()* and record elapsed time by using *time.time()*. The results are illustrated in Fig. 7b. We use 5 seconds as the interval time between two transactions of eight types. The confirmation time for the transactions varies due to network delay and consensus mechanism. To reduce the transaction confirmation, we can decrease the consensus time in the blockchain network according to application's requirements. Last, we test the network latency in accessing data, i.e., the time elapsed from a CPI's submitting an access transaction to receiving a download link L from a blockchain node. The average network latency ranges from 0.58s to 0.68s as illustrated in Fig. 7c. The network delay is directly related to the blockchain network and the underlying consensus mechanism, which is controllable by means of reliable infrastructure and network.

7.5 Performance Comparison With Existing Work

The basic parameter settings of the experiments are the same for comparison. We notice that not all the schemes share the same system model or procedure which makes it not easy to comparison, still we manage to find the common entities to compare their computational costs and communication overheads. For the entities or functions which do not exist in a scheme, we mark its value as "n/a". The comparison results are listed in Tables 4 and 5.

For DiaLog [25], we choose an AES key of 128-bit, an RSA key of 1024-bit, and a message of 160-bit. Data providers/requesters register with a Central Communication Unit



(a) Consensus time (b) Transaction confirmation time (c) Waiting time of accessing evidence

Fig. 7. Network delay.

TABLE 5
Communication Overhead (Unit: KBytes)

Phase	Entity Registration				Data Uploading		Data Accessing		Data Retrieval			Result Reporting		Traitor Tracing
Entity	DP	DU	AU	BN	DP	BN	DU	BN	AU	DU	BN	DU	BN	CPI[1]
DiaLOG [25]	0.05	0.05	0.05	n/a	n/a		n/a		0.28	0.05	n/a	n/a		n/a
DFAV [26]	n/a				0.31	n/a	n/a		0.25	0.03	n/a	n/a		n/a
B4F [6]	n/a	n/a	0.09	n/a	0.20	0.16	n/a		n/a			n/a		n/a
BB-VDF [12][2]	n/a				0.7	0.09	0.19	4.78[3]	0.32	7.34	7.34	0.13	0.13	n/a
Eunomia	0.58	0.88	0.58	0.58	2.45	2.45	2.95	2.95	17.42	0.56	0.64	0.18	0.18	73.8[4]

[1]*Traitor tracing happens when a crime police investigator submits leaked data to the crime police department.* [2]*In BB-VDF, we use SHA256 as* H_0*, the attributes number in the access policy* $l = 5$*.* [3]*There are two parts for the 4.78 KB: 2.39 KB for court and 2.39 KB for BN.* [4]*The length of an original data block is 36.1 KB, and the one of an embedded data block is 73.8 KB.*

(CCU) to obtain symmetric keys. AES encryption is used to encrypt data. RSA is used to generate a signature when the CCU returns encrypted data.

Data providers in DFAV [26] computed a ciphertext of data by using AES-128 and a hash of data. A ciphertext of the encryption key is also generated by using RSA and the public key of the database. As a result, the above two centralized solutions have lower execution time and overhead.

B4F [6] adopts the IEEE 1609.2 standard to guarantee anonymity and perform encryption and signing by using the Elliptic Curve Integrate Encrypt Scheme (ECIES). Certification authorities manage the users' identities. Although the three schemes have a lower cost and overhead, they do not address most of the privacy and security concerns vital to VDF.

Data providers in BB-VDF [12] encrypt data by using a distributed key policy attribute-based encryption with a partially hidden access structures scheme. Data users can request data from the data storage by obtaining a decryption key from the authorities.

In entity registration, unlike other schemes, all four entities in Eunomia need to register. This is a foundation for privacy and security measures. In data uploading, the blockchain nodes in Eunomia have to verify the identity of data providers. In data accessing, Eunomia has a higher time cost than BB-VDF for verifying zero-knowledge proofs sent by data users. In data retrieval, we can see a higher execution time from Eunomia for embedding watermark and generating multiple zero-knowledge proofs. In traitor tracing, the CPD in our scheme tracks the secret key of the traitor.

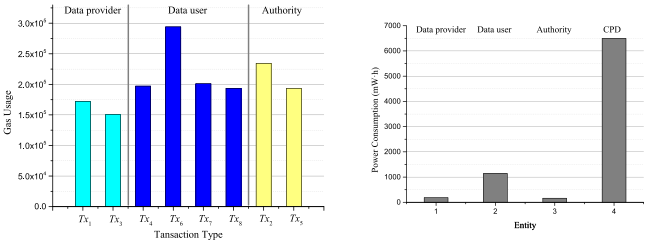
7.6 Monetary Cost and Power Consumption

Gas is originally designed to defend Denial-of-Service attack in Ethereum. It is achieved by costing a pre-defined amount of gas for running a smart contract in which each operation has a fixed gas cost. gasLimit and gasPrice are specified by a transaction sender. gasLimit is the maximum amount of gas used in executing this transaction. gasPrice is the number of Ethereum coin (called Ether) that the sender is willing to spend per unit of gas.

We measure the gas cost of eight types of transactions. Specifically, they are classified into three sets: Tx_1 and Tx_3 for data provider; Tx_4, Tx_6, Tx_7, and Tx_8 for data user; and Tx_2 and Tx_5 for authority (blockchain node). From Fig. 8a, the average gas usage for Tx_1 is about 172,156. The gas price in the Eunomia blockchain is 1 Gwei (0.000000001 Ether), at the exchange rate of 597.36 USD per Ether at the time of writing (December 3, 2020). Each Tx_1 costs about 0.00017 Ether (or 0.10 USD). The gas difference of different transaction exists because gas varies when the transaction items are different. Even when two transactions are of the same type, the anonymous credential and timestamp vary for different data providers in Tx_1.

Now we measure the power consumption of implementing the system by using Windows powercfg command[11]. To improve accuracy, we close unnecessary programs and run the computers in the battery mode. Specifically, we measure the power consumption for: data provider (register and upload), data user (register, access, retrieve, analyze (only

11. https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/powercfg-command-line-options



(a) Monetary cost (b) Power consumption

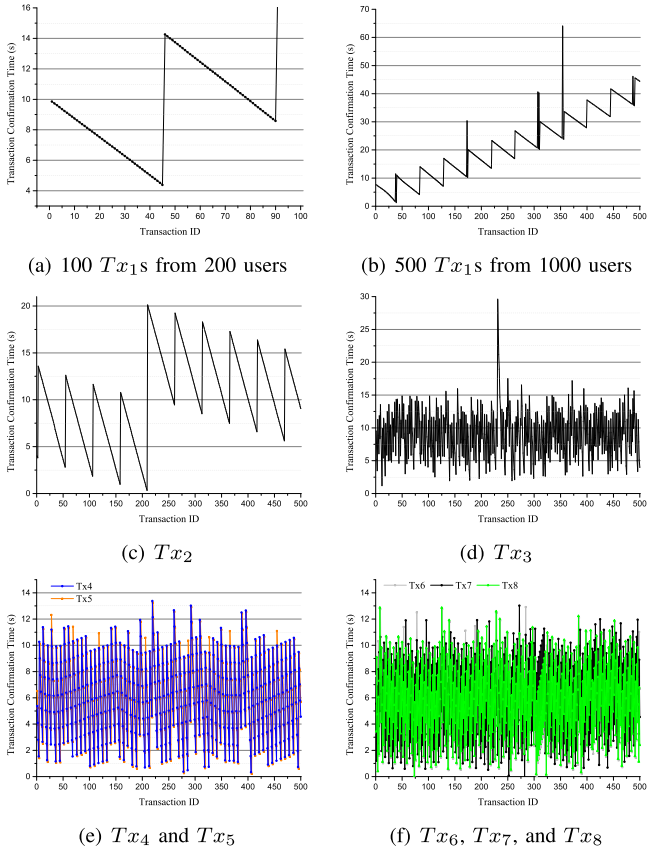
Fig. 8. Monetary cost and power consumption of Eunomia.

upload Tx_6), report, and close), blockchain node (verify and consensus), and CPD (retrieve). For example, the power consumption of a blockchain node is measured during one block time, i.e., from receiving transactions from 100 users to creating a new block. As shown in Fig. 8b, a data provider only consumes 181 mW·h and a data user consumes 1141.5 mW·h. It costs a node 160 mW·h and costs the CPD 6493.5 mW·h in one data retrieval. The monetary cost and power consumption show that it only costs the data providers and data users a little money and energy to send transactions to the Ethereum blockchain, which does not hinder them from using the blockchain-based VDF system. Especially, if the forensic system has its own consortium blockchain, the cost will be further reduced.

7.7 Scalability

Scalability refers to how the blockchain scales with the increase in the network size [40]. We create 1000 user accounts using geth client. Each account receives 1000 ether from one pre-funded miner account. Since web3py is a none thread-safe library, we write transaction scripts in a loop. To reduce the time interval between two transactions, we prepare all needed parameters and sources, and execute transaction submitting operations in the loop. We measure the Transaction Confirmation Time (TCT) for the eight types of transactions. In Fig. 9a, the number of users is 200 (100 data providers/users). The transaction time interval of Tx_1 is 0.1s. The TCT for Tx_1 follows a periodic pattern. Within one period, all transactions are packed into one block. The line decreases linearly because the transactions sent first are packed first. For different periods, the TCT for later periods are longer because some transactions are sent in the previous periods and transactions that are not packed will wait for the following blocks when the total gas of transactions in the current block has reached the block gas limit. When the number of Tx_1s is 500 in Fig. 9b, the TCT will increase. Some "odd" points exist because the corresponding transactions contain longer messages which are cut in line by shorter transactions. In Fig. 9c, the time interval of requests is set to 0.2s, and we can also see a periodic pattern in the TCT of Tx_2s. For data uploading in Fig. 9d, we randomly draw the time interval of Tx_3s from 1s to 5s, the resulting curve does not show a clear pattern that validates the random sampling. Tx_4s (access, 1s interval time) and Tx_5s (permit) are mixed together. One Tx_4 is followed by one Tx_5. Fig. 9e shows that Tx_4 and Tx_5 share the same pattern of TCT because they come in pairs. The TCT of Tx_6s, Tx_7s, and Tx_8s in Fig. 9f is analogous to Fig. 9d due to the randomness in transaction generation. The average TCT for Tx_4, ..., Tx_8 is lower than 10s, because their time interval is longer than previous settings, so they are packed faster.

The scalability experimental results show that the total time needed for reaching consensus of a new block is stable



(a) 100 Tx_1s from 200 users (b) 500 Tx_1s from 1000 users

(c) Tx_2 (d) Tx_3

(e) Tx_4 and Tx_5 (f) Tx_6, Tx_7, and Tx_8

Fig. 9. Scalability of Eunomia.

regardless of the network size. Therefore, we can support more data providers and data users to participate in the system. If the network scales, data providers/users can mark their uploading/accessing transactions by an indicator to emphasize the urgency. Then the blockchain nodes will pick transactions based on the indicator. In addition, authorities can decrease block interval and increase block size according to the increase in number of transactions.

8 Discussions

8.1 Uploading Falsified Information

Since Eunomia collects data using a crowdsourcing method, it is possible that some data provider (e.g., a threatened civilian, a collusive gangster) uploads falsified information to the system. The falsified information could be classified into two types: forged and fake. The first one refers to blood planted at crime scenes and an edited photo/video. The impacts of this misconduct could be reduced by forensics professionals with a set of skills acquired through years of experience, and professional techniques specially designed to detect any trace of forgery. For example, there are image forgery detection [41], video forgery detection [42]. Such techniques can be integrated into the Eunomia app as the first line of defense in data uploading. The second one refers to some made-up testimony. To mitigate the effects of the fake evidence, we ask the data providers to offer more convincing and detectable evidence. Moreover, data providers will be asked to testify at court given that this is a requirement of current justice systems including USA [43], Canada [44], and UK [45].

9 Conclusion

In this paper, we have presented an anonymous and secure vehicular digital forensics scheme Eunomia based on blockchain, smart contract, and cryptographic primitives. It provides anonymity, unlinkability, data confidentiality, authentication, fine-grained access control, accountability, and traceability. With Eunomia, data providers can readily participate in the forensic procedure without privacy concerns. Data users can request the collected data to conduct an investigation while not disclosing their ongoing tasks. Malicious insiders cannot tamper with evidence easily since all the data related operations are documented in the shared ledger. Traitors who leak data will be traced given that their personal identifier is embedded into their requested data.

Acknowledgements

This work was carried out during the tenure of an ERCIM 'Alain Bensoussan' Fellowship Programme granted to Meng Li. Thank Professor Zijian Zhang from BIT for discussions about technical challenges and formal security definitions. Thank Professor Jianbing Ni from Queen's University for discussions about technical challenges.

References

[1] L. Zhu, M. Li, Z. Zhang, and Z. Qin, "ASAP: An anonymous smart-parking and payment scheme in vehicular networks," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 4, pp. 703–715, Jul./Aug. 2020. doi: 10.1109/TDSC.2018.2850780.

[2] M. Li, L. Zhu, and X. Lin, "Efficient and privacy-preserving carpooling using blockchain-assisted vehicular fog computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4573–4584, Jun. 2019, doi: 10.1109/JIOT.2018.2868076.

[3] 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.*, to be published, doi: 10.1109/TDSC.2020.3017534.

[4] S. Savage, "Lawful device access without mass surveillance risk: A technical design discussion," in *Proc. 25th ACM Conf. Comput. Commun. Secur.*, Toronto, ON, Canada, Oct. 2018, pp. 1761–1774.

[5] J. Lacroix, K. El-Khatib, and Rajen Akalu, "Vehicular digital forensics: What does my vehicle know about me?," in *Proc. 6th ACM Symp. Develop. Anal. Intell. Veh. Netw. Appl.*, New York, NY, USA, Nov. 2016, pp. 59–66.

[6] 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.

[7] Joshua A. Kroll, E. W. Felten, and D. Boneh, "Secure protocols for accountable warrant execution," Apr. 2014. [Online]. Available: https://www.cs.princeton.edu/ felten/warrant-paper.pdf

[8] J. Frankle, S. Park, D. Shaar, S. Goldwasser, and D. Weitzner, "Practical accountability of secret processes," in *Proc. 27th USENIX Secur. Symp.*, Baltimore, MD, USA, Aug. 2018, pp. 657–674

[9] A. Kiayias and Q. Tang, "Traitor deterring schemes: Using bitcoin as collateral for digital content," in *Proc. 25th ACM Conf. Comput. Commun. Secur.*, Denver, CO, USA, Oct. 2015, pp. 231–242.

[10] M. Li, L. Zhu, and X. Lin, "Privacy-preserving traffic monitoring with false report filtering via fog-assisted vehicular crowdsensing," *IEEE Trans. Serv. Comput.*, to be published, doi: 10.1109/TSC.2019.2903060.

[11] M. Li, L. Zhu, Z. Zhang, C. Lal, M. Conti, and M. Alazab, "Anonymous and verifiable reputation system for E-commerce platforms based on blockchain," *IEEE Trans. Netw. Serv. Manage.*, to be published, doi: 10.1109/TNSM.2021.3098439.

[12] M. Li, J. Weng, J.-N. Liu, X. Lin, and C. Obimbo, "Towards vehicular digital forensics from decentralized trust: An accountable, privacy-preservation, and secure realization," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2021.3116957.

[13] Do Warrants Expire In California? Accessed: Oct. 24, 2021. [Online]. Available: https://www.adanieldefense.com/do-warrants-expire-in-california

[14] Z. A. Baig *et al.*, "Future challenges for smart cities: Cyber-security and digital forensics," *Digit. Investigation*, vol. 22. pp. 3–13, 2017.

[15] C. Cai, Y. Zheng, Y. Du, Z. Qin, and C. Wang, "Towards private, robust, and verifiable crowdsensing systems via public blockchains," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 4, pp. 1893–1907, Jul./Aug. 2021.

[16] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[17] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Yellow Paper, 2014. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf

[18] C. Garman, M. Green, and I. Miers, "Decentralized anonymous credentials," in *Proc. 21st Netw. Distrib. Syst. Secur. Symp.*, San Diego, CA, USA, Feb. 2014, pp. 1–15.

[19] A. Nieto, R. Rios, and J. Lopez, "Digital witness and privacy in IoT: Anonymous witnessing approach," in *Proc. 16th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Sydney, NSW, Australia, Aug. 2017, pp. 642–649.

[20] S. Agrawal and M. Chase, "FAME: Fast attribute-based message encryption," in *Proc. 24th ACM Conf. Comput. Commun. Secur.*, Dallas, TX, USA, Oct. 2017, pp. 665–682.

[21] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs short proofs for confidential transactions and more," in *Proc. 39th IEEE Symp. Secur. Privacy*, San Francisco, CA, USA, May 2018, pp. 315–334.

[22] Alibaba Cloud - Object Storage Service. Accessed: Oct. 24, 2021. [Online]. Available: https://www.alibabacloud.com/product/oss?spm=a3c0i.11270126.9836925980.17.15a225b3knZQ2a

[23] G. G. Richard III and V. Roussev, "Next-generation digital forensics," *Commun. ACM*, vol. 49, no. 2, pp. 76–81, 2006.

[24] S. L. Garfinkel, "Digital forensics research: The next 10 years," *Digit. Investigation*, vol. 7, pp. S64–S73, 2010.

[25] H. Mansor, K. Markantonakis, R. N. Akram, K. Mayes, and I. Gurulian, "Log your car: The non-invasive vehicle forensics," in *Proc. 15th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun.* Tianjin, China, Aug. 2016, pp. 974–982.

[26] X. Feng, E. S. Dawam, and S. Amin, "A new digital forensics model of smart city automated vehicles," in *Proc. 10th IEEE Int. Conf. Internet Things*, Exeter, U.K., Jun. 2017, pp. 274–279.

[27] C. Huang, R. Lu, and K.-K. R. Choo, "Vehicular fog computing: Architecture, use case, and security and forensic challenges," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 105–111, Nov. 2017.

[28] N.-A. Le-Khac, D. Jacobs, J. Nijhoff, K. Bertens, and K.-K. R. Choo, "Smart vehicle forensics: Challenges and case study," *Future Gener. Comput. Syst.*, vol. 109, pp. 500–510, 2020.

[29] C. Dai, X. Xiao, Y. Ding, L. Xiao, Y. Tang, and S. Zhou, "Learning based security for VANET with blockchain," in *Proc. 16th IEEE Int. Conf. Commun. Syst.*, Chengdu, China, Dec. 2018, pp. 210–215.

[30] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity and a methodology of cryptographic protocol design," in *Proc. 27th Annu. Symp. Found. Comput. Sci.*, Toronto, ON, Canada, Oct. 1986, pp. 174–187.

[31] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Proc. 3th Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA, Aug. 1986, pp. 186–194.

[32] T. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Proc. 8th Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA, Aug. 1991, pp. 129–140.

[33] V. Goyal, O. Pandey, A. Sahai, B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," *Proc. 13rd ACM Conf. Comput. Commun. Secur.*, Alexandria, VA, USA, Oct. 2006, pp. 89–98.

[34] S. F. Kelty, R. Julian, and J. Robertson, "Professionalism in crime scene examination: The seven key attributes of top crime scene examiners," *Forensic Sci. Policy Manage.*, vol. 2, pp. 175–186, 2011.

[35] J. G. Chamani, D. Papadopoulos, D. Papadopoulos, and R. Jalili, "New constructions for forward and backward private symmetric searchable encryption," in *Proc. 25th ACM Conf. Comput. Commun. Secur.*, Toronto, ON, Canada, Oct. 2018, pp. 1038–1055.

[36] J. Katz and Y. Lindell, *Introduction to Modern Cryptography* 2nd ed. Boca Raton, FL, USA: CRC Press, 2015.

[37] A. B. Lewko and B. Waters, "Unbounded HIBE and attribute-based encryption," in *Proc. 28th Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, Tallinn, Estonia, May 2011, pp. 547–567.

[38] C. Priebe, K. Vaswani, and M. Costa, "EnclaveDB: A secure database using SGX," in *Proc. 39th IEEE Symp. Secur. Privacy*, San Franciso, CA, USA, May 2018, pp. 264–278.
[39] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.
[40] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proc. 23th ACM Conf. Comput. Commun. Secur.*, Vienna, Austria, Oct. 2016, pp. 17–30.
[41] Y. Quan and C.-T. Li, "On addressing the impact of ISO speed upon PRNU and forgery detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 190–202, Jul. 2021.
[42] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial video forgery detection network," in *Proc. 10th IEEE Int. Workshop Inf. Forensics Secur.*, Hong Kong, China, Dec. 2018, pp. 1–7.
[43] United States Department of Justice, "Tips for testifying in federal court United States Department of Justice." Accessed: Oct. 24, 2021. [Online]. Available: https://www.justice.gov/usao-edwi/victim-witness-assistance/coming-court/tips-testifying
[44] Depratment of Justice, "The Trial." Accessed: Oct. 24, 2021. [Online]. Available: https://www.justice.gc.ca/eng/cj-jp/victims-victimes/court-tribunaux/trial-proces.html
[45] GOV.UK, "Going to court to give evidence as a victim or witness." Accessed: Oct. 24, 2021. [Online]. Available: https://www.gov.uk/going-to-court-victim-witness

Meng Li (Member, IEEE) received the BE degree in information security from the Hefei University of Technology in 2010, the MS and PhD degrees in computer science and technology from the Beijing Institute of Technology in 2013 and 2019, respectively. He is currently an associate professor and the assistant to dean with the School of Computer Science and Information Engineering, Hefei University of Technology, China. He is also a postdoctoral fellow with the Department of Mathematics, University of Padua, Italy, where he is with the SPRITZ Research Group. He was sponsored by the ERCIM 'Alain Bensoussan' Fellowship Programme in October 2019 to conduct postdoctoral research at CNR, Italy. From 2017 to 2018, he was sponsored by the China Scholarship Council (CSC) to study with Broadband Communications Research (BBCR) Lab, University of Waterloo and Wilfrid Laurier University. He has authored or coauthored more than 40 papers in journals and conference, including the *IEEE Transactions on Dependable and Secure Computing*, *IEEE Transactions on Services Computing*, *IEEE Transactions on Network and Service Management*, MobiCom, ICICS, SecureComm, TrustCom, and IPCCC in his research area. His research interests include security and privacy, fairness, vehicular networks, applied cryptography, edge computing, and blockchain.

Yifei Chen (Student Member, IEEE) received the BE degree from the Hefei University of Technology, Hefei, China, in 2019. He is currently working toward the MS degree with the School of Computer Science and Information Engineering, Hefei University of Technology. His research interests include applied cryptography, security and privacy, vehicular networks, blockchain, and SGX.

Chhagan Lal (Senior Member, IEEE) received the PhD degree in computer science and engineering from the Malaviya National Institute of Technology, Jaipur, India, in 2014. He is currently a postdoctoral research fellow with Delft University of Technology, The Netherlands. He was a postdoctoral fellow with the Department of Mathematics, University of Padua, Italy, where he was part of the SPRITZ Research Group. His research interests include applications of blockchain technologies, security in software-defined networking, and Internet of Things networks.

Mauro Conti (Fellow, IEEE) received the PhD degree from the Sapienza University of Rome, Italy, in 2009. He was a postdoctoral researcher with Vrije Universiteit Amsterdam, The Netherlands. He is currently a full professor with the University of Padua, Italy. He is also affiliated with TU Delft and the University of Washington, Seattle. In 2011, he joined as an assistant professor with the University of Padua, where he became an associate professor in 2015 and a full professor in 2018. He is visiting researcher with GMU, UCLA, UCI, TU Darmstadt, UF, and FIU. He has authored or coauthored more than 400 papers in topmost international peer-reviewed journals and conferences. His research is also funded by companies, including Cisco, Intel, and Huawei. His research focuses on Security and Privacy. He is an area editor-in-chief of the *IEEE Communications Surveys and Tutorials*, an associate editor for several journals, including *the IEEE Communications Surveys and Tutorials*, *IEEE Transactions on Dependable and Secure Computing*, *IEEE Transactions on Information Forensics and Security*, and the *IEEE Transactions on Network and Service Management*. He was the program chair for TRUST 2015, ICISS 2016, WiSec 2017, ACNS 2020, and the general chair for SecureComm 2012, SACMAT 2013, CANS 2021, and ACNS 2022. He is a senior member of the ACM. He is a member of the Blockchain Expert Panel of the Italian Government. He is Fellow of the Young Academy of Europe. He was the recipient of Marie Curie Fellowship 2012 by the European Commission and with a Fellowship by the German DAAD 2013.

Mamoun Alazab (Senior Member, IEEE) received the PhD degree in computer science from the Federation University of Australia, School of Science, Information Technology and Engineering. He is currently an associate professor with the College of Engineering, IT and Environment, Charles Darwin University, Australia. He is a cybersecurity researcher and practitioner with industry and academic experience. He has authored or coauthored more than 150 research papers. His research interests include cybersecurity and digital forensics of computer systems with a focus on cybercrime detection and prevention including cyber terrorism and cyber warfare. He delivered many invited and keynote speeches, 24 events in 2019 alone. He convened and chaired more than 50 conferences and workshops. He is closely with government and industry on many projects, including Northern Territory (NT) Department of Information and Corporate Services, IBM, Trend Micro, the Australian Federal Police (AFP), the Australian Communications and Media Authority (ACMA), Westpac, United Nations Office on Drugs and Crime (UNODC), and the Attorney Generals Department. He is the founding chair of IEEE Northern Territory (NT) Subsection.

Donghui Hu (Member, IEEE) received the BS degree from Anhui Normal University, China, in 1995, the MS degree from the University of Science and Technology of China, China, in 2004, and the PhD degree in information security from Wuhan University, China, 2010. From 2013 to 2014, he was a visiting researcher with UNCC. He is currently a Professor with the School of Computer Science and Information Engineering, Hefei University of Technology. His research interests include information trustworthiness evaluation and privacy protection.