BB-VDF: Enabling Accountability and Fine-grained Access Control for Vehicular Digital Forensics through Blockchain

Journal:	*Transactions on Dependable and Secure Computing*
Manuscript ID	TDSC-2019-07-0383
Manuscript Type:	Regular Paper
Keywords:	Vehicular digital forensics, blockchain, accountability, fine-grained, privacy preservation

SCHOLARONE™
Manuscripts

BB-VDF: Enabling Accountability and Fine-grained Access Control for Vehicular Digital Forensics through Blockchain

Ming Li, Jian Weng*, *Member, IEEE*, Jia-Nan Liu, *Student Member, IEEE*, Xiaodong Lin, *Fellow, IEEE*, Charlie Obimbo

Abstract—With the increasing number of traffic accidents and terrorist attacks by modern vehicles, vehicular digital forensics (VDF) has gained significant attention in identifying and determining evidences from the related digital devices. Ensuring the law enforcement agency to accurately integrate various kinds of data is a crucial point to determine the facts. However, malicious attackers or semi-honest participants may undermine the digital forensic procedures. Enabling accountability and privacy preservation while providing secure fine-grained data access control in VDF is a non-trivial challenge. To mitigate this issue, in this paper, we propose a blockchain-based scheme for VDF named BB-VDF, in which the accountable protocols and privacy preservation methods are constructed. The desirable security properties and fine-grained data access control are achieved based on the customized smart contracts and cryptographic constructions. Specifically, we design novel smart contracts that model the forensics procedures as a finite state machine, which guarantees accountability that each participant performs auditable cooperation under tamper-resistant and traceable transactions. Furthermore, we design a distributed key-policy attribute based encryption scheme with partially hidden access structures to realize the secure fine-grained forensics data access control. Systematic security analysis and extensive experimental results show the feasibility and practicability of the proposed BB-VDF scheme.

Index Terms—Vehicular digital forensics, blockchain, accountability, fine-grained, privacy preservation.

1 Introduction

The functionalities of vehicles have been strengthened tremendously with the increasing in-vehicle sensors, control units, and communication methods, such as Electronic Control Unit (ECU), Bluetooth, and Wi-Fi [1]. According to statistics from *Ford Motor Company* [2], a modern-day vehicle has approximately 50-70 computers, which enables it to be an important source of digital data (e.g., locations where doors are open/close, when the tires are pumped and lights are on/off). These wealth of sensing and operation data make vehicles more intelligent and smarter, which will prompt the prosperity of the autonomous driving industry effectively in the near future [3].

However, as everything has its two sides, the increasing smart vehicles also bring us tons of security issues. Using vehicles as weapons to conduct terrorist attacks are not rare and have caused tremendous damages and losses to our society [4], [5]. Specially, vehicle ramming attacks (VRA), which is a typical used attack, refer to malicious actions of deliberately using a vehicle to ram to a building or a crowd of people. For example, in July 14, 2016, a 20-ton rental truck rammed into the crowd who were watching a firework display in Nice, France, which killed 86 people and wounded more than 450 people [6]. Since it is incredibly easy to get one from rental companies, many terrorist attackers choose a rental car as the criminal weapon [7]. In other words, launching a VRA requires minimal capability while has the prospects to cause catastrophic disasters to the society. It has spread like a virus in recent years. According to a *cnn.com* report on terrorist attacks caused by vehicle [8], at least 7 major attacks happened in 2017, which led to at least 37 people killed and hundreds of pedestrians injured.

For these types of VRAs, a forensics investigation specialized for vehicle can be conducted to analyze the suspicious behaviors and collect evidences, which has been called *Vehicular Digital Forensics* [9] (VDF) (also called "Vehicle Forensics" [10]). VDF has gained considerable attention both in academic and industrial area since a vast amount of data being collected by in-vehicle computers. It can help law enforcement agency to detect a potential VRA by identifying suspicious activities. In particular, this field becomes more significant with the forthcoming of car-sharing and self-driving cars, which are the way of the future (*vehicle* and *car* are used interchangeably in this paper) [3]. However, it also brings a burning question: who is at fault if a self-driving car gets involved in a fatal accident, the driver or the car manufacturer who develops the self-driving algorithms? If it is in the latter case, the manufacturer could be sued for an unprecedented amount of money for a lost life, and eventually go out of business. Thus, it has become crucial to have a forensically sound way for authorities to investigate car accident in the era of autonomous driving.

As for VRA, the law enforcement agency may prevent it from happening beforehand if enough data is obtained for

- *M. Li, J. Weng, and J. Liu are with the College of Information Science and Technology and the College of Cyber Security, Jinan University, Guangzhou 510632, China. Jian Weng is the corresponding author. E-mail: cryptjweng@gmail.com, limjnu@gmail.com.*
- *X. Lin and C. Obimbo are with the School of Computer Science, University of Guelph, Guelph ON N1G 2W1, Canada. E-mail: xlin08@uoguelph.ca, cobimbo@uoguelph.ca.*

VDF. For instance, a rental company can detect that someone has difficulty in explaining the purposes of renting a car. Combining with other related data, such as traffic management center to report that the car parks in a specific area for several days without any reasonable explanation, the law enforcement agency may confirm that it is a potential threat to public safety in that area [11]. It is obvious that a single data source is not enough for the analysis of suspicious behaviors, thus the comprehensive historical data from the car and related other data sources need be obtained by the investigators. Unfortunately, it is not easy to conduct a vehicle forensics investigation due to the existence of several security issues. To illustrate, take the following hypothetical scenario as an example:

Example: *Carl is a terrorist attacker and intends to launch a VRA. He rents a car as the criminal tool. When Carl appears in a specific place, a pedestrian Alice discovers that Carl's behaviors are suspicious. Alice calls the policeman Bob immediately and tells him the necessary information. Consequently, Bob launches a VDF to investigate the case. He first applies for a warrant from the court. After being authorized, Bob uses the permitted warrant to request the historic data on that car from related parties, such as the corresponding rental company, traffic management center, and maintenance service provider.*

Actually, there are several security and privacy issues in the above example that may have adverse impacts on the implementation of VDF: 1) the detailed contents of the warrant may be leaked to Carl by malicious external attackers, which alerts Carl and cause him to abandon the attacks temporarily; 2) Bob may abuse his power to acquire more data that are unrelated to the car from the parities, or even tamper the collected evidences; and 3) there may exist malicious insiders in the related parties who modify the historical data before presenting to Bob, or claim the historical data has been lost, which apparently violates the digital chain of custody [12]. Apart from these, there exist other problems which are harmful to the normative VDF procedures. Specifically, as vehicles become smarter and more complicated than before, it is hard for the law enforcement agency to get forensics data due to the lack of specialized tools, but to appeal for technical help from a commercial party. However, it may bring the potential threat of privacy leakage. Besides, since the court has released large number of warrants accumulatively, she/he may forget to trace the warrants' states, which allows semi-honest investigators to still use these warrants to obtain secret data (even though they are expired) [13]. It is not an easy work for the court to trace the states of all released warrants in reality.

We note that while some existing schemes have been proposed to solve parts of the issues [10], [13]–[18], most of them are for different applications, and under different system models or security threats. Specifically, the public should be able to audit the process of VDF while preserving the privacy, which assures the accountability and legitimacy of the forensics process without the misuse or abuse of power. In addition, the forensics data should be securely obtained by the law enforcement agency with fine-grained access control, nothing more and nothing less. It is non-trivial to consider the above security issues and challenges in VDF scenario simultaneously. The closest to our work is [14] which is the first research that proposes a framework on integrating different parties' data to conduct the vehicle forensics based on blockchain. However, they do not focus on resolving the challenges that the confidentiality of the warrants should be preserved during the forensics process (especially for the terrorist attacks), and the law enforcement agency or other parties may behave dishonestly.

To address these challenges, we explore the potential of blockchain and smart contract, and take advantage of cryptographic primitives to design a blockchain-based VDF scheme with accountability, privacy preservation, and fine-grained data access control. We construct a permissioned blockchain to integrate data from different parties (including rental companies, traffic management center, car manufacturers, car maintenance centers) to assist the law enforcement agency to accomplish an investigation. The vehicle forensics process is modeled as a finite-state machine (FSM) in smart contracts, which enables the involved parties to cooperate and act legally under the supervision of the public. Further, to generalize the forensic data access control in a fine-grained manner, we propose a distributed key-policy attribute based encryption (KP-ABE) scheme with partially hidden access structures to protect the privacy of the attributes (i.e., the sensitive warrant information) and eliminate single point of failure/compromise (SPoF/C) issues on secret key management. Specifically, our **contributions** can be summarized as follows:

- We propose a blockchain-based VDF scheme with accountability and privacy preservation named BB-VDF, in which the privacy of the warrant details and forensic data are preserved by leveraging the customized cryptographic algorithms. The distributed KP-ABE with partially hidden access structures scheme is designed to ensure secure fine-grained access control on forensic data for VDF.
- We design a set of smart contracts that model the vehicle forensics process as an FSM. The FSM follows the *verification-then-forwarding* model that any state transition needs to satisfy the pre-designed conditions and should be confirmed by the majority of blockchain nodes without privacy breaches, which enables the VDF processes to be publicly verifiable and traceable. It can ensure that regulatory requirements are followed properly to avoid inadmissible evidence (or forensic data).
- We implement a prototype of the proposed scheme and deploy it to the Ethereum public test network *Rinkeby*. Extensive experimental results indicate the feasibility and practicability of the proposed BB-VDF scheme.

Paper Structure. The remainder of the paper is organized as follows. In the next Section, we introduce the background and related preliminaries. In Section 3, we formalize the system model, threat model, and the security goals. In Section 4, we present the proposed concrete scheme. The proof and security analysis are given in Section 5. In Section 6, we present the experiments and evaluation results. The related works are given in Section 7. Finally, we give the conclusion in Section 8.

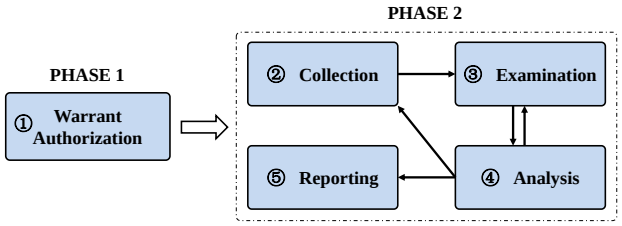


Fig. 1. The workflow of digital forensics.

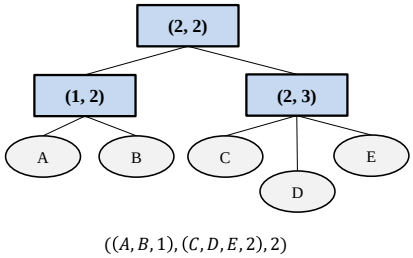


Fig. 2. An example of access policy in KP-ABE scheme.

2 Background and Preliminaries

2.1 Digital Forensics

As illustrated in Fig. 1, a typical digital forensics can be depicted in two main phases: 1) the first phase is the warrant authorization that law enforcement agency (e.g., the policeman) requests a valid authorization from the court before accessing data of any individual entity [16]. The warrant contains a signature from the court to permit the law enforcement agency to conduct the investigation; and 2) The second phase is on data processing. It contains four steps: *collection*, *examination*, *analysis*, and *reporting* [19]. Specifically, *collection* is the process of evidences collection that aims to gather sufficient data from the software system (e.g., mobile App) or hardware devices (e.g., physical RAM and SD card). These data and devices should be stored with due care to protect the integrity and confidentiality. *Examination* is to perform search of the data that is related with the respondent or the suspected crime. *Analysis* aims to conduct more systematical and professional analysis on the collected data or devices. *Reporting* is responsible for providing the final investigation reports on the results of the previous process.

2.2 Blockchain and Smart Contract

Most recently, the blockchain technology has been employed in many applications, such as financial services [20], healthcare [21], internet-of-things (IoT) [22], [23] and crowdsourcing [24], [25]. It is essentially a distributed ledger that maintained by a number of network nodes (also called blockchain nodes) [26]. Blockchain nodes may be mutual distrust while can still reach an agreement based on the consensus protocol, e.g., proof of work (PoW) or proof of stake (PoS). More preciously, the blockchain is composed of a series of consecutive *blocks*, i.e, an ordered hash chain. Each *block* contains a number of *transactions*. Its security assurance is based on the cryptographic primitives that ensure the transmissions of digital currency or status transitions among different entities in a secure way.

Particularly, the review of main features on blockchain can be listed as follows: 1) *Complete Decentralization*: it is based on distributed P2P network that many untrusted nodes can achieve fair data exchange without reliance on a central party. 2) *Correct Execution*: blockchain is a global computer that each blockchain node can trace and verify the correctness of the data computation. 3) *Tamper-resistance*: the data (i.e., blocks and transactions) are tamper-resistant since they are organized as the special data structure (Merkle tree and hash chain).

And also, smart contracts is used to construct the decentralized application (DApp) [27], which facilitates the process of an application to be executed automatically on blockchain technology. People can participate in a DApp by providing valid inputs to execute a function in smart contract. Such function execution corresponds to a transaction on the chain.

2.3 Cryptography Algorithms

In our constructions, we make use of the following cryptographic algorithms as building blocks to achieve the accountability and fine-grained access control.

Bilinear Pairing: Let $\mathbb{G}_1$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p. g is a random generator of $\mathbb{G}_1$. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$ be a computable bilinear pairing with the follow properties:

- Bilinearity: for all $g \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p$, we have $e(g^a, g^b) = e(g^b, g^a) = e(g, g)^{ab}$.
- Non-degeneracy: $e(g, g) \neq 1$.

Distributed Key Generation (DKG): DKG is one of the components in (t, n)-threshold cryptosystem [28]. It allows several n parties to collectively generate a key pair (i.e., public key and private key) without letting any single party to reconstruct or store the secret key. Besides, it also does not rely on any trusted party to achieve t-secure, which means that the protocol is secure if no more than $t + 1$ parties are broken. Further, Gennaro *et al.* [29] improved the security of DKG protocol with the uniform randomness property. By running the DKG protocol proposed in [29], each honest party will hold a share α_i of a secret key α. For any each set $\mathcal{N}$ of $t + 1$ correct shares, $\alpha = \sum_{i \in \mathcal{N}} \lambda_i \cdot \alpha_i$, where λ_i are Lagrange interpolation coefficients for set $\mathcal{N}$. Specially, the t-secure DKG protocol will always satisfy the following *correctness* and *secrecy* properties:

- *Correctness:* Any subsets of $t + 1$ shares define the same privacy key α ($\alpha \in \mathbb{Z}_p$) and all parties share the same public key $y = g^\alpha$.
- *Secrecy:* There is no information learned on x expect for the implication of value $y = g^\alpha$.

Key Police Attribute-Based Encryption (KP-ABE): KP-ABE scheme is a type of public key encryption. It allows user to encrypt and decrypt data based on attributions. Compared with the identity-based encryption (IBE) scheme, KP-ABE scheme is more suitable to support fine-grained access control policy. As shown in Fig. 2, the leaf nodes refer to attributes and non-leaf nodes refer to threshold

gates. It denotes a comprehensible access structure, namely, $((A, B, 1), (C, D, E, 2), 2)$. Generally, KP-ABE consists of the following four algorithms [30]:

-Setup$(1^\eta) \rightarrow (PK, MSK)$. The setup algorithm takes a security parameter η as the input and outputs the public parameters PK and a master secret key MSK. It chooses a bilinear group $\mathbb{G}_1$ of prime order p, and $\alpha \in \mathbb{Z}_p$. g is a random generator of $\mathbb{G}_1$, x refers to the attributions. $H(x)$ is a hash function: $H : \{0,1\}^* \rightarrow \mathbb{G}_1$. PK and MSK can be presented as follows:

$$PK = (\mathbb{G}, p, g, e(g,g)^\alpha, H(x)), MSK = \{\alpha\}. \quad (1)$$

-Encrypt$(PK, M, S) \rightarrow CT$. The encryption algorithm takes the public parameters PK, a set of attributes S and a message M as the inputs. It selects a random number $s \in \mathbb{Z}_p$ outputs a ciphertext $CT = \{S, C, \hat{C}, \{C_x\}_{x \in S}\}$ as follows:

$$C = M \cdot e(g,g)^{\alpha s}, \hat{C} = g^s, \{C_x = h_x^s\}_{x \in S}. \quad (2)$$

-KeyGen$(PK, MSK, \mathbb{A}) \rightarrow SK$. The key generation algorithm takes the public parameters PK, the master secret key MSK and an access structure $\mathbb{A}$ as the inputs and outputs the private key SK, $\mathbb{A} = (W, \rho)$ is an LSSS access structure. W is an $l \times n$ matrix, ρ is the function that maps the rows of W to attributes [31]. Γ is the set of distinct attributes the appear in W, $\Gamma = \{d : \exists i \in [1, l], \rho(i) = d\}$. The algorithm selects a random vector $\overrightarrow{v} = (\alpha, y_2, ..., y_n)$. For $i = 1$ to l, it calculates $\mu_i = \overrightarrow{v} \cdot W_i$. Then, it chooses random $r_1, ..., r_l \in \mathbb{Z}_p$ and computes SK as follows:

$$PK, (D_1 = g^{\mu_1} \cdot h_{\rho(1)}^{r_1}, R_1 = g^{r_1}, \forall d \in \Gamma/\rho(1), Q_{1,d} = h_d^{r_1}),$$
$$..., (D_l = g^{\mu_l} \cdot h_{\rho(l)}^{r_l}, R_\ell = g^{r_l}, \forall d \in \Gamma/\rho(l), Q_{l,d} = h_d^{r_l}). \quad (3)$$

-Decrypt$(PK, SK, CT) \rightarrow M$. The decryption algorithm takes as input the public parameters PK, a private key SK, and a ciphertext CT associated with a set of attributes S. If the set S of attributes satisfies the access structure $\mathbb{A}$, then the algorithm will decrypt the ciphertexts and return a message M. Let $I \subseteq \{1, ..., l\}$ be a set of indices, $\Delta = \{x : \exists i \in I, \rho(i) = x\}$, and $\{\omega_i\}_{i \in I} \in \mathbb{Z}_p$. The algorithm decrypts the cyphertext as follows:

$$L = \prod_{x \in \Delta} C_x = \prod_{x \in \Delta} h_x^s.$$
$$e(g,g)^{\alpha s} = e(\hat{C}, \prod_{i \in I} \hat{D}_i^{\omega_i}) / e(\prod_{i \in I} \hat{R}_i^{\omega_i}, L) \quad (4)$$

3 System and Threat Models and Security Goals

In this section, we formally present the blockchain-based VDF framework and system model. Then, we give our security assumptions and discuss the threat model. Finally, the security goals are clearly defined.

TABLE 1
The notations of explanation.

Notation	Explanation
$\mathcal{L}$	The law enforcement agency.
$\mathcal{C}$	The court.
$\mathcal{B}$	The blockchain platform.
$\mathcal{D}$	The distributed data storage system.
$(\mathcal{A}_1, ..., \mathcal{A}_n)$	The n decryption authorities.
$(\mathcal{S}_1, ..., \mathcal{S}_m)$	The data sources.
PK, MSK	The public parameters and master secret key.
$id, type, t$	The identifier, type and timestamp of forensics data.
K_e^p, K_e^s	The entity's public key and private key.
$M_1 \| \| M_2$	The concatenation of message M_1 and M_2.
H_0, H_1, H_2	Three non-cryptographic hash functions.
$\mathsf{Enc}_{sk}(M)$	The symmetric encryption on message M with private symmetric key sk, e.g., AES.
$r', r_x, r_{x,y}$	The generated random numbers.
$\mathbb{A} = (W, \rho)$	The access structure in KP-ABE.
Ti	The timestamp in the transaction.

3.1 System Model

As illustrated in Fig. 3, there exist six parties involved in our proposed scheme: Data Sources, Law Enforcement Agency, Court, Decryption Authorities, Blockchain Platform, Distributed Data Storage. Each party has a corresponding key pair (i.e., public key and private key). The notations that will be utilized in this paper are presented in TABLE 1.

- Data Sources: identified by $\mathcal{S} = \{\mathcal{S}_1, ..., \mathcal{S}_i, ..., \mathcal{S}_m\}$, refer to the different entities who can provide the forensics data, including the automotive vehicles equipped with digital devices (e.g., electronic control units (ECUs) and GPS systems), rental companies, traffic management center, car manufacturers, and car maintenance centers. $\mathcal{S}$ will generate the necessary data which are helpful for VDF and store them to the distributed data storage based on forensics-by-design paradigm [32]. We assume that each vehicle has the On Board Units (OBUs) that can be used to communicate with the roadside units or other vehicles with the Dedicated Short Range Communications (DSRC) protocol [33]. The data sources can submit the forensics data according to the similar algorithms in our design.
- Law Enforcement Agency: identified by $\mathcal{L}$, refers to the investigator (e.g., the policeman) who is responsible for launching a digital forensics investigation. $\mathcal{L}$ is assumed to have some professional skills (including software and hardware skills) to acquire data from $\mathcal{S}$ and the data storage system.
- Court: identified by $\mathcal{C}$, refers to the official judges who can approve $\mathcal{L}$'s request to conduct an investigation on a vehicle according to the specified legal standard.
- Decryption Authorities, identified by $\mathcal{A} = \{\mathcal{A}_1, ..., \mathcal{A}_j, ..., \mathcal{A}_n\}$, refer to the entities who jointly maintain a master secret key by using the DKG protocol. They provide the shares to allow $\mathcal{L}$ to recover the decryption key if she/he has an authorized warrant. In particular, $\mathcal{L}$ needs to obtain at least $t + 1$ shares to decrypt the data downloaded from the data storage. $\mathcal{A}$ can be the established

organizations in real-world deployment, e.g., the government departments.

- Blockchain Platform: identified by $\mathcal{B}$, recognized as a permissioned blockchain that is maintained by multiple blockchain nodes. There are several roles that can act as the blockchain nodes in BB-VDF, such as the court, the law enforcement agency, and the decryption authorities. Other parties are allowed to join in this ecosystem with the permission. Specially, the state of a warrant is recorded in $\mathcal{B}$, which enables the public to audit the validity and legitimacy of the investigation.
- Distributed Data Storage, identified by $\mathcal{D}$, refers to the data storage system that stores the related forensics data. Our scheme adopts the distributed data storage techniques that can be utilized in our design (e.g., S3). The data is encrypted in $\mathcal{D}$ that if $\mathcal{L}$ intends to retrieve data, she/he needs to be granted with the authorized access by $\mathcal{C}$ and $\mathcal{A}$ to obtain a decryption key.

Inspired by [14], we assume that a *forensics daemon* runs inside the OBU and will submit the data which are related with VDF to $\mathcal{B}$ and $\mathcal{D}$ periodically. More preciously, if there exists a suspicious behavior or an accident that needs to be investigated, $\mathcal{L}$ first applies for a valid warrant from $\mathcal{C}$. If $\mathcal{C}$ permits, she/he will issue a warrant cryptographically signed by her/his secret key to allow $\mathcal{L}$ to acquire data. After that, $\mathcal{L}$ first requires a decryption key from $\mathcal{A}$ using the authorized warrant, and then collect data from $\mathcal{D}$. Each step during the investigation is required to submit a transaction to prompt the state machine transition in smart contracts, which will be described in subsection 4.1.

The underlying $\mathcal{B}$ is built atop of the existing permissioned blockchain, e.g., Ethereum. Transaction fee is not considered in our scheme, which is the incentive problem. As widely known, the Ethereum blockchain can support Turing-complete smart contract (e.g., *solidity*) which is extremely useful for constructing the state machine and accomplishing auditable forensics investigations. To construct a secure blockchain platform, we design that some special blockchain nodes (e.g., the $\mathcal{C}$) have higher weight to maintain the security of $\mathcal{B}$ than other nodes. *Proof-of-stake* (PoS) based blockchain platform can be used to support this design [34].

An embedded hardware-security-module (HSM) based scheme [35] is adopted in our scheme that any data request through vehicle's ECUs should be authorized, which ensures ECUs with secure communications for on-board system. When it is necessary to collect directly from the vehicle (e.g., traffic accidents), the forensics daemon within OBU communicates with different ECUs through CAN bus requires access authorization, which guarantees the security of data collection. In particular, the data is encrypted using hybrid encryption method before being sent to $\mathcal{D}$. The data is encrypted using symmetric encryption algorithm and the symmetric key is encrypted based on the customized KP-ABE scheme. In addition to the vehicles, the other data sources will generate some operations and maintenance data on a specific vehicle which are helpful for the forensics. These parties are required to upload the data to $\mathcal{D}$ periodically, and $\mathcal{L}$ can not be allowed to require data directly from them. Particularly, their data encryption and decryption algorithms are in consistent with the vehicles, so other parties' forensics data retrieval will not be emphasized in the concrete scheme.

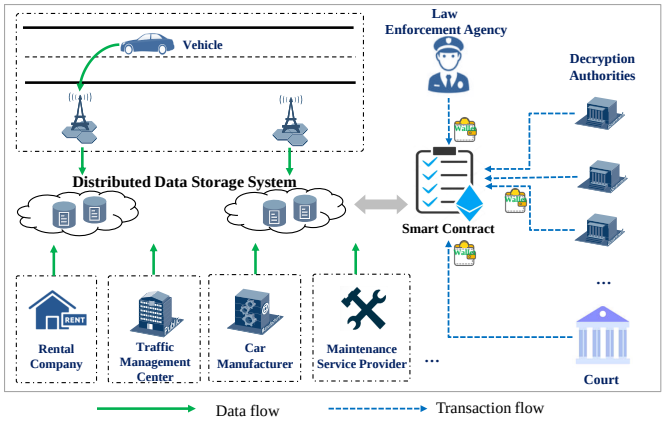


Fig. 3. System model.

3.2 Threat Model

Without loss of generality, the security of $\mathcal{B}$ follows the *majority-honest-assumption* that if most of the blockchain nodes are honest, $\mathcal{B}$ will be run accurately and the smart contract will be accurately executed with valid inputs [36].

Moreover, we assume the forensics daemon is run in a trusted hardware that malicious attackers can not disturb the normal running, or tamper the content of the collected data. Each data submission will be signed with vehicle's secret key. Similar with the prior schemes on digital forensics, all parties involved in our scheme are assumed with bounded computation, and most of them are honest and perform their duties properly. However, there still exist some malicious insiders or external attackers who may undermine the forensics investigation. We assume that part of $\mathcal{A}$ may be compromised or dishonest, as long as no more than a pre-defined number (depending on the protocol design in real-world). $\mathcal{L}$, $\mathcal{C}$, and honest $\mathcal{A}$ will not collude with each other during the investigation. Specifically, we mainly focus on the following security threats:

Threat 1: Malicious Law Enforcement Agency. $\mathcal{L}$ usually belongs to a trusted institution in reality. However, there exist malicious insiders who are in pursuit of individual profit do not follow the standardized digital forensics process. Thus, we consider that $\mathcal{L}$ may be untrustworthy in the threat model. In particular, a warrant issued by $\mathcal{C}$ designates the detailed investigation information (e.g., the data type that $\mathcal{L}$ can acquire), which specifies that $\mathcal{L}$ can not acquire more data than the designated during the investigations. However, malicious $\mathcal{L}$ (denote as $\mathcal{L}^*$) may attempt to 1) acquire more data without explicit approval, 2) take advantage of an expired warrant to obtain access to unauthorized data, or 3) alter or forge the collected evidences before presenting to $\mathcal{C}$. Furthermore, potential external attackers may attempt to compromise $\mathcal{L}$ and impersonate an authorized $\mathcal{L}$ to access the data or conduct further attacks.

Threat 2: Honest-but-Curious Court. $\mathcal{C}$ is a trusted institution who will comply with the designated protocols to make a judgement. Similar with [16], $\mathcal{C}$ can not learn about the details of data records besides the related forensics data. However, there may exist malicious insiders who are curious about the detailed data records. In addition, $\mathcal{C}$ may forget to track the states of issued warrants, which allows $\mathcal{L}$ to acquire data by the expired warrants.

Threat 3: Untrustworthy Authorities. None of $\mathcal{A}$ can learn about the details of an warrant and data records, especially for the terrorist attacks investigation. Generally, if given an authorized warrant, $\mathcal{A}$ will provide the correct secret shares honestly. However, partial compromised or internal malicious $\mathcal{A}$ (denotes as $\mathcal{A}^*$) may exist and attempt to get the details of a warrant, e.g., the identity of the suspicious vehicle. Then, they may collude with a suspicious vehicle, which allows the vehicle to change her/his behavior. Besides, $\mathcal{A}^*$ may attempt to learn about the plaintext of the data records.

3.3 Security Goals

In order to enable accountability and fine-grained access control for VDF and resist the aforementioned security threats, the proposed scheme should achieve the following high-level security goals:

3.3.1 Accountability

Accountability in digital forensics can be considered as a secure assurance that the related parties (including $\mathcal{C}, \mathcal{L}$, and $\mathcal{A}$) will not misuse or abuse their powers during the investigation forensics. Our proposed scheme will achieve accountability from the following aspects:

Complete Process Audit: To ensure the legitimacy of an investigation for a digital forensics, the crucial process nodes should be identified and each process node needs to be audited by the public with the blockchain system. Specially, each process node transformation should have a digital signature and follows the legal standards. Take the warrant authorization for an example, before $\mathcal{L}$ requires a decryption key from $\mathcal{A}$, she/he needs to apply for a digital warrant signed by $\mathcal{C}$, which is accomplished by submitting a transaction to $\mathcal{B}$. The blockchain nodes will check the validity of the request, and prompt the process to the next stage if the signature is valid.

Public Verifiability: The proposed scheme should be able to prevent unauthorized actions or behaviors during the process of investigation. Specially, each process node should be publicly scrutiny, which means the involved parties should be accountable for the investigation and prevented from abusing or misusing their granted power.

3.3.2 Privacy Preservation

The proposed should preserve the secrecy of the warrant and forensics data as follows:

Confidentiality of Warrant: The detailed information of a warrant should be temporarily protected to prevent any unauthorized entity from learning about it for a period of time, especially in terms of VRA investigation. We design that $\mathcal{L}$ and $\mathcal{C}$ know the warrant details, while the unauthorized parties can only know the metadata of the warrant, e.g., the short description and its hash value.

Confidentiality of Forensics Data: The content of forensics data should be protected except the authorized $\mathcal{L}$, and if the forensics data is generated during the production, renting and maintenance process, then also the car manufacturers, rental companies and maintenance centers, respectively.

3.3.3 Data Security

The proposed scheme should be able to achieve the following data security goals:

Availability: The scheme should ensure the service and data availability that resist against DDoS attacks and SPoF/C.

Integrity: To ensure the validity and legitimacy of the evidence, the scheme should be able to protect the integrity of the data (e.g., $\mathcal{V}$), which is also the requirement of digital chain of custody that the digital evidence presented in the $\mathcal{C}$ should be consistent with the original generated data without tampering or corrupting. Note that our scheme mainly focuses on data integrity protection after the data has been generated.

Unforgeability: The data records or the intermediate generated parameters that will be used in the forensics can not be forged by any malicious users.

4 Proposed Scheme

In this section, we present the design of the proposed scheme with the high-level security goals in the VDF, and address the security threats described in the above section. Section 4.1 outlines the overview of the proposed scheme and depicts the blockchain based forensics state machine. Section 4.2 introduces the concrete scheme which explores cryptographic primitives to achieve accountability, privacy preservation, data security.

4.1 Overview of Our Scheme

In the concrete scheme, we use the vehicles (i.e., $\mathcal{S}$) as illustration to denote how the forensics data are generated and collected. Our proposed scheme can be extended to support other data sources with a little change. As mentioned before, a forensics daemon runs in OBU to collect the related data that might be helpful for forensics (e.g., the latest vehicle operations) based on different sensors. The forensics daemon submits these data to $\mathcal{D}$ periodically when there is a good network connection. Specially, we design that the hash value of the forensics data is submitted to $\mathcal{B}$ periodically with a short interval, while the encrypted data records are stored off-chain, the real-time requirements are not so high, thus it can be stored with a more high interval.

Specifically, $\mathcal{L}$ generates a warrant information which contains the metadata and detailed data. The metadata can be published in $\mathcal{B}$ and the detailed data which specifies the forensics data range (i.e, the attributions) is kept secret from any unauthorized parties. After the warrant being authorized by $\mathcal{C}$, $\mathcal{L}$ can conduct the digital forensics on the specific object. We mainly focus on the processes of warrant authorization, collection and reporting (i.e., ①, ②, ⑤ as in Fig. 1). During the collection phase, $\mathcal{L}$ applies for a secret key from $\mathcal{A}$ and collects the forensics data on two aspects: 1)

the vehicle itself and 2) data storage $\mathcal{D}$. To acquire real-time data from the vehicle, $\mathcal{L}$ can use the authorized secret key to generate a signature, the forensics daemon will verify the validation of the signature and send a diagnostic command to each ECU, and $\mathcal{L}$ will get the ciphertexts. On the other hand, to collect a vehicle's historical data from $\mathcal{D}$, $\mathcal{L}$ uses the secret key to find the *download link*, and downloads the corresponding data from $\mathcal{D}$. As for the data analysis phase, our scheme follows the idea of block4forensics [14] that integrates different kinds of data to conduct the analysis. Finally, the time-scheduling service in smart contract is designed for the reporting.

State Machine: An FSM is constructed to depict the vehicle forensics in smart contracts as illustrated in Fig. 4. According to the practical scenario, we model the process nodes as some states and represent them in smart contracts. The state machine essentially represents a life cycle for a warrant. Each state denotes a global process node of the warrant. In particular, we design more concrete states than the states in the general workflow as in Fig. 1. The states follow *verification-then-forwarding* model that any state transition should be combined with a digital signature in a transaction and verified by the multiple blockchain nodes.

Fig. 4 shows different states that a warrant can be in and how the state transits. There are eight states in FSM: Warrant Request, Warrant Authorization, Shares Retrieval, Data Collection, Data Examination, Data Analysis, Forensics Report, and Completed. The initial state Warrant Request refers to $\mathcal{L}$ initializes to request a warrant. We define different "events" that trigger the state transition. Only after $\mathcal{C}$ permits the request can the following FSM execution happens, otherwise the state will be transited to Completed by $\mathcal{C}$. The inputs for different states derived from different parties. If $\mathcal{L}$ finds more data needs to be collected when the state is in Data Analysis, she/he needs to send an order modification request to $\mathcal{C}$ and the state is transferred into the initial Warrant Request. More precisely, a state σ_1 that is transited into a new state σ_2 according to a transaction TX is denoted as: $\sigma_1 \xrightarrow{TX} \sigma_2$.

Furthermore, we design a timer to schedule some specific tasks in FSM based on the time-scheduling service [37]. If one of state transitions fails, e.g., a specified condition can not be satisfied by the off-chain parties for a pre-defined time interval, then the corresponding parties are to be informed or the forensics will be aborted. As the above mentioned, $\mathcal{C}$ may forget to track the states of the warrants which are actually expired, while other parties may not realize the expiration and still maintain cooperation, which is apparently unsatisfied with the legal process. To address this issue, our scheme ensures that each involved party has a clear understanding on the state of the warrant, and knows when do they cooperate for the investigation. Specifically, a service called Ethereum Alarm Clock (EAC) is adopted to provide the time-scheduling service [38]. It allows us to schedule transactions to be executed after a certain time. In doing so, some of the state transitions are accomplished by specifying the pre-prescribed actions which follow the standard investigation of VDF. For example, if a state transaction of warrant is not satisfied for a pre-defined time, the EAC service could automatically transit the state to Completed.

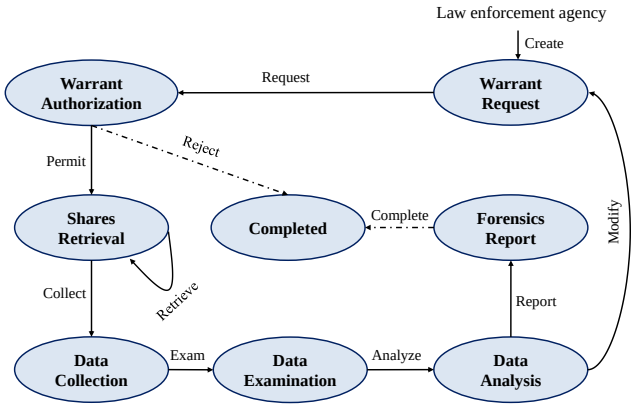


Fig. 4. The state machine model for vehicle digital forensics.

4.2 Concrete Scheme

In this subsection, we present our concrete scheme by leveraging the blockchain and cryptographic tools.

4.2.1 System Initialization and Configuration

In BB-VDF, the court $\mathcal{C}$ is responsible for initializing the whole system in the system setup phase. More precisely, given the security parameters η, $\mathcal{C}$ generates the public parameters. Let $\mathbb{G}_1$ be a bilinear group with prime order $p \in \Theta(2^\eta)$. g is a random generator of $\mathbb{G}_1$. Besides, let H_0, H_1, H_2 be three cryptographic hash functions: $H_0 : \{0,1\}^* \rightarrow \mathbb{Z}_p$, $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_T \rightarrow \{0,1\}^\kappa$, where κ is the length of the symmetric encryption key, e.g., 128 bits.

To resist against SPoF/C and prevent single party from abusing the power, our scheme employs the generating phase of secure DKG protocol as in [29]. By doing so, each authority $\mathcal{A}_j$ holds a secret share α_j and jointly maintains a master secret key α with other authorities. After that, a set $\mathcal{N}$ of $t+1$ authorities publish their partial public parameters $v_j = e(g,g)^{\alpha_j}$ and generate the system public parameters as:

$$\prod_{j \in \mathcal{N}} \left(e(g,g)^{\alpha_j} \right)^{\lambda_j} = e(g,g)^{\sum_{j \in \mathcal{N}} \lambda_j \cdot \alpha_j} = e(g,g)^{\alpha}. \quad (5)$$

Then, the trusted authority specifies the master public parameters PK as $PK = (\mathbb{G}, p, g, e(g,g)^\alpha, H_0, H_1, H_2)$. The master private key is $MSK = \{\alpha\}$ which is not controlled by any party. Besides, involved parties have their own key pairs (public key and private key). Their public keys are authenticated by $\mathcal{C}$ and published in the blockchain that anyone can verify the validity of a signature [1].

4.2.2 Digital Forensics Data Generation

During the data generation phase, the parties, including vehicles, car manufactures, and maintenance centers, are required to periodically submit the related data to $\mathcal{D}$. The period can be flexibly set, such as one hour or an half hour for the consideration of real-time forensics. Assume that a forensics data submitted by $\mathcal{S}_i$ is denoted as $d_{id,type,t}$, where id refers to the identity of $\mathcal{S}_i$ (i.e., $K^p_{\mathcal{S}_i}$), $type$ refers to the data type (e.g., steering wheel, brake, seat belt), and

1. The identity management and authentication is not depicted here, which is not the key point in this paper.

t refers to the timestamp. For the simplify of analysis, we assume that the vehicles collect data at an integer time. These variables are essentially defined as *attributes*, such as tagging a data $d_{id,type,t}$ with the metadata ("*id*: Alice", "*type*: steering wheel", "*t*: 2019/07/22 22:00").

In terms of the vehicle forensics data generation, the forensics daemon in $\mathcal{S}_i$ interacts with the ECUs and collects data $\beta_{t_1} = \{d_{id,type_1,t_1}, ..., d_{id,type_m,t_1}\}$ at time t_1. These data are encrypted with hybrid encryption method and sent to the off-chain $\mathcal{D}$. Meanwhile, $\mathcal{S}_i$ submits a transaction to $\mathcal{B}$, which stores the metadata (e.g., hash value, timestamp) on-chain. In doing so, the data integrity can be verified with the tamper-resistant transaction records in the blockchain. It is worth noting that $\mathcal{S}_i$ does not need to submit a transaction for each $d_{id,type,t}$, she/he can construct a Merkle Hash tree based on the consecutive several data (e.g., $\beta_{t_1 \sim t_\varsigma} = \{\beta_{t_1}, ..., \beta_{t_\varsigma}\}$) and store the Merkle root value to the blockchain. During the data analysis phase, $\mathcal{L}$ can verify the integrity of a specific data using the Merkle tree proof verification algorithm [39]. Specifically, the process of the digital forensics data generation can be described with the following steps:

- To encrypt $d_{id,type,t_1}$, the forensics daemon randomly generates two number $s_1 \in \mathbb{Z}_p, \varepsilon_1 \in \mathbb{G}_T$ and computes the hash value $k_1 = H_2(\varepsilon_1)$. Then, $\mathcal{S}_i$ uses k_1 to encrypt the data $d_{id,type,t_1}$ with symmetric encryption algorithm (e.g., AES-128) and computes the values as follows:

$$\begin{aligned} &k_1 = H_2(\varepsilon_1), \hat{C} = g^{s_1}, M_{id,type,t} = Enc_{k_1}(d_{id,type,t_1}), \\ &H_{M_{id,type,t_1}} = H_0(M_{id,type,t_1}), C = \varepsilon_1 \cdot e(g,g)^{\alpha s_1}, \\ &\{C_x = H_1(x)^{s_1}\}_{x \in \{0,1\}^*}, \theta_1 = H_0(id||type||t_1), \end{aligned} \tag{6}$$

  where θ_1 is defined as the corresponding *download link* of $d_{id,type,t_1}$. $\{\theta_1 : (M_{id,type,t_1}, C, \hat{C}, \{C_x\}_{x \in \{0,1\}^*})\}$ are stored to $\mathcal{D}$. Given a download link, private information retrieval (PIR) technique can be adopted when retrieve the data from $\mathcal{D}$, which protects the privacy of the attributes without exposing others [40].
- After collecting ς period data, the forensics daemon computes the Merkle root value $root$, where $root = MerkleRoot(H_{M_{id,type_1,t_1}}, ..., H_{M_{id,type_m,t_\varsigma}})$ by mapping the consecutive $m \times \varsigma$ ciphertexts to a group element $root$ [39].
- Vehicle $\mathcal{V}_i$ prepares and signs the transaction with the private key:

$$TX^1_{\mathcal{V}_i} = [root, (\theta_1, ..., \theta_{m \times \varsigma}), Ti]_{K^s_{\mathcal{V}_i}}, \tag{7}$$

  where Ti refers to the generation timestamp of the transaction.

4.2.3 Warrant Request and Authorization

In case of accidents or suspicious behaviors, $\mathcal{L}$ will send a request req to $\mathcal{C}$ to apply for a warrant before conducting to the substantive investigation. req refers to the detailed information of the request which includes the description of metadata and attribution data (i.e., target identity, data type, and time). The metadata is opened for public verification while the attributions are kept secret. The request req can contain multiple vehicles, different data types and timestamp, which can be achieved by constructing a comprehensive access policy $\mathbb{A}$. Let $\mathbb{A} = (W, \rho)$, W be an $\ell \times u$ matrix, and ρ be the function that associates rows of W to the attributes. Particularly, the multiple decryption authorities can jointly generate a private decryption key for the access policy. For instance, our scheme allows $\mathcal{L}$ to decrypt the ciphertexts with attribution sets satisfying the access policy: ("*id*: David" OR "*id*: Carl") AND (2019/06/21 22:00 $\leq$"*time*"$\leq$ 2019/06/23 22:00).

Specifically, $\mathcal{L}$ submits a transaction with des to the smart contract to create a new state machine life cycle. The initial state is set as Warrant Request as in Fig. 4. $\mathcal{L}$ privately sends the structure of access policy $\mathbb{A}$ to $\mathcal{C}$ through a secure channel. If the request is valid, $\mathcal{C}$ will sign on the request and generate some intermediate parameters which are used for recovering the decryption key. Otherwise, the investigation will be rejected and the state transits into Completed. The processes of warrant request and authorization can be depicted as follows:

- $\mathcal{L}$ sends a warrant request req which contains the access policy $\mathbb{A}$ and metadata des to $\mathcal{C}$, and submits a transaction to create a new warrant state machine *FSM* in the smart contract. $H_{req} = H_0(req)$ is computed as the unique identifier of the warrant in $\mathcal{B}$. The public can audit the process of a legitimate forensics investigation based on the identifier.

$$TX^2_{\mathcal{L}} = [des, H_{req}, Ti]_{K^s_{\mathcal{L}}}. \tag{8}$$

  An FSM instance is created in smart contract as: $* \xrightarrow{TX^2_{\mathcal{L}}}$ Warrant Request.
- $\mathcal{C}$ receives the structure $\mathbb{A}$ and evaluates whether to approve the investigation request. If no, $\mathcal{C}$ submits a transaction to terminate the FSM instance (i.e., prompts the FSM to Completed). Otherwise, $\mathcal{C}$ selects ℓ sets of random numbers $r_1, ..., r_\ell \leftarrow \mathbb{Z}_p$ and generates the following values:

$$\begin{aligned} &\xi_1 = (H_1(\rho(1))^{r_1}, g^{r_1}, \forall d \in \Gamma / \rho(1), H_1(d)^{r_1}), \\ &... \\ &\xi_\ell = (H_1(\rho(\ell))^{r_\ell}, g^{r_\ell}, \forall d \in \Gamma / \rho(\ell), H_1(d)^{r_\ell}), \\ &res = (\xi_1, \xi_2, ..., \xi_\ell), \end{aligned} \tag{9}$$

  where res is the response message that will be sent back to $\mathcal{L}$ through the secure channel.
- $\mathcal{C}$ submits a transaction with a digital signature on the value of $(H_1(\rho(1))^{r_1}, ..., H_1(\rho(\ell))^{r_\ell})$. In doing so, $\mathcal{A}$ or the public can confirm that the decryption key request from $\mathcal{L}$ is authorized by $\mathcal{C}$ without power abusing:

$$TX^3_{\mathcal{C}} = [H_{req}, H_0(res), H_1(\rho(1))^{r_1}, ..., H_1(\rho(\ell))^{r_\ell}, Ti]_{K^s_{\mathcal{C}}}. \tag{10}$$

If $\mathcal{C}$ permits the request and the blockchain has confirmed the validity of the transaction, the FSM instance will transit into the new state as: Warrant Request $\xrightarrow{TX^3_{\mathcal{C}}}$ Warrant Authorization.

4.2.4 Auditable Data Collection

To securely retrieve a private decryption key that satisfies the access structure $\mathbb{A}$, $\mathcal{L}$ blinds the values res and sends them to different authorities and aims to collect enough

shares to recover the decryption key. Each authority measures whether to provide the share to $\mathcal{L}$ based on the clues and judgement.

The master secret key is maintained by n decryption authorities without being recovered by any single authority. To recover the decryption key, $\mathcal{L}$ requires a set $\mathcal{N}$ of $t+1$ decryption key shares from $\mathcal{A}$. Each authority $\mathcal{A}_j$ will blind it with some random numbers to protect the privacy and submit a transaction to the blockchain. In doing so, any authorities' behavior is recorded in transaction without repudiation for the auditability of forensics. Note that before providing the share to $\mathcal{L}$, $\mathcal{A}_j$ will verify the state of the state machine on whether is Share Retieval and req is a legitimate investigation.

To protect the privacy of investigation, we design that the attribution values $\rho(1),...,\rho(\ell)$ are not revealed to any party in the customized KP-ABE scheme. Let $\Gamma=\{d:\exists i\in[1,\ell],\rho(i)=d\}$. $\mathcal{L}$ chooses a set of random numbers $r_{\epsilon_1,d}\in\mathbb{Z}_p,\epsilon_1\in[1,\ell]$ and computes the blinded values on res as follows:

$$\begin{aligned}&\varphi_1=(H_1(\rho(1))^{r_1},g^{r_1r_{1,1}},\forall d\in\Gamma/\rho(1),H_1(d)^{r_1r_{1,d}}),\\&...\\&\varphi_\ell=(H_1(\rho(\ell))^{r_\ell},g^{r_\ell r_{\ell,1}},\forall d\in\Gamma/\rho(\ell),H_1(d)^{r_\ell r_{\ell,d}}),\end{aligned}\tag{11}$$

Then, $\mathcal{L}$ sends $\{\varphi_1,...,\varphi_\ell\}$ to each authority. Note that the values $\{g^{r_x},\forall d\in\Gamma/\rho(x),H_0(\rho(d))^{r_x}\},x\in[1,\ell]$ have been blinded with random $r_{\epsilon_1,d}$. In doing so, $\mathcal{A}_j$ can not launch testing attacks to learn about the values of the attributes. Meanwhile, $\mathcal{L}$ sends a transaction to the blockchain to apply for the decryption key in $\mathcal{B}$.

$$\begin{aligned}&TX^4_{\mathcal{L}}=[H_{req},H_1(\rho(1))^{r_1},...,H_1(\rho(\ell))^{r_\ell},\\&H_0(\varphi_1),...,H_0(\varphi_\ell),Ti]_{K^s_{\mathcal{L}}}.\end{aligned}\tag{12}$$

The state machine is transited as Warrant Authorization $\xrightarrow{TX^4_{\mathcal{L}}}$ Shares Retrieval. After that, each authority will verify whether the private decryption key request is corresponding to the authorized request by $\mathcal{C}$, which is easy to check by the published transactions $TX^3_{\mathcal{C}}$ and $TX^5_{\mathcal{L}}$. If $\mathcal{A}_j$ confirms the request is legitimate, she/he will join the data collection phase to recover the forensics data. As for $\mathcal{A}_j$, she/he uses the secret share α_j which is a share of the master secret key $MSK=\{\alpha\}$ to compute SK_j, a share of the decryption key SK (note that $\alpha=\sum_{j\in\mathcal{N}}\lambda_j\alpha_j$). Since α is not reconstructed explicitly by any single authority, $\mathcal{A}_j$ can not compute $\{g^{\mu_1},...,g^{\mu_\ell}\}$ directly as in [30], but she/he can use the share α_j to compute the intermediate parameters, i.e., $\{g^{\mu_{1,j}},...,g^{\mu_{\ell,j}}\}$, where $g^{\mu_{a,j}}=g^{W_a\cdot\vec{v_j}}=g^{W_a\cdot(\alpha_j,y_2,...,y_n)}$, $a\in[1,\ell]$. Specially, to prevent $\mathcal{L}$ or the public from knowing the value of $g^{\mu_{a,j}}$, $\mathcal{A}_j$ chooses a random number $r'_j\in\mathbb{Z}_p$ and computes the secret share $\widetilde{SK_j}=(\widetilde{\varphi_{1,j}},...,\widetilde{\varphi_{\ell,j}})$ as follows:

$$\begin{aligned}&(g^{\mu_{1,j}}H_1(\rho(1))^{r_1r'_j},g^{r_1r_{1,1}r'_j},\forall d\in\Gamma/\rho(1),H_1(d)^{r_1r_{1,d}r'_j}),\\&...\\&(g^{\mu_{\ell,j}}H_1(\rho(\ell))^{r_\ell r'_j},g^{r_\ell r_{\ell,1}r'_j},\forall d\in\Gamma/\rho(\ell),H_1(d)^{r_\ell r_{\ell,d}r'_j}),\end{aligned}\tag{13}$$

$\mathcal{A}_j$ sends $\widetilde{SK_j}$ to $\mathcal{L}$. In particular, to allow the public to verify that the provided shares are indeed computed according to the authorized warrant, we can let the authority to approve that $(\widetilde{\varphi_{1,j}},...,\widetilde{\varphi_{\ell,j}})$ are indeed computed based on the authorized parameters $res=(\xi_1,...,\xi_\ell)$ using zero knowledge proof technique [41]. In addition, each authority is required to submit a transaction to prove the validity of share-providing. By taking $\mathcal{A}_j$ as an instance, the transaction is shown as follows:

$$TX^5_{\mathcal{A}_j}=[H_{req},g^{\mu_{1,j}}H_1(\rho(1))^{r_1r'_j},...,g^{\mu_{\ell,j}}H_1(\rho(\ell))^{r_\ell r'_j},Ti]_{K^s_{\mathcal{A}_j}}\tag{14}$$

Note that $\mathcal{L}$ can recover the decryption key SK if she/he collects any set $\mathcal{N}$ of $t+1$ correct shares $\{\widetilde{SK_\tau}\}_{\tau\in\mathcal{N}}$. Specifically, $\mathcal{L}$ uses $r_{\epsilon_1,d}$ to recover the final secret shares by multiplying exponentially with $1/r_{\epsilon_1,d}$. For example, to recover SK_1, $\mathcal{L}$ respectively multiplies exponentially with $1/r_{1,1},...,1/r_{1,\ell}$ on $(g^{r_1r_{1,1}r'_1},\forall d\in\Gamma/\rho(1),H_1(\rho(d))^{r_1r_{1,j}r'_1})$ and gets $(g^{r_1r'_1},\forall d\in\Gamma/\rho(1),H_1(\rho(d))^{r_1r'_1})$. $\mathcal{A}_j$ does not need to reveal the secret share α_j, while $\mathcal{L}$ can still figure out the blinded shares without learning about any private information. $\mathcal{L}$ restructures SK according to secure DKG protocol as follows:

$$\begin{aligned}&\{PK,(g^{\mu_1}\cdot H_1(\rho(1)))^{r_1\sum_{\tau\in\mathcal{N}}r'_\tau\cdot\lambda_\tau},g^{r_1\sum_{\tau\in\mathcal{N}}r'_\tau\cdot\lambda_\tau},\forall d\in\Gamma/\rho(1),\\&H_1(d)^{r_1\sum_{\tau\in\mathcal{N}}r'_\tau\cdot\lambda_\tau}),...,(g^{\mu_\ell}\cdot H_1(\rho(\ell)))^{r_\ell\sum_{\tau\in\mathcal{N}}r'_\tau\cdot\lambda_\tau},\\&g^{r_1\sum_{\tau\in\mathcal{N}}r'_\tau\cdot\lambda_\tau},\forall d\in\Gamma/\rho(\ell),H_1(d)^{r_1\sum_{\tau\in\mathcal{N}}r'_\tau\cdot\lambda_\tau})\}\end{aligned}\tag{15}$$

Provided that more than t authorities have provided the shares, the FSM instance is automatically transited into the new state as: Shares Retrieval $\xrightarrow{\{TX^5_{\mathcal{A}_\tau}\}_{\tau\in\mathcal{N}}}$ Data Collection.

After restructuring the secret key SK, $\mathcal{L}$ retrieves the related data from $\mathcal{D}$ according to the attributes. Note that, based on the PIR protocol, the process of data retrieval does not expose the information whose data have been downloaded. Then, $\mathcal{L}$ computes the data decryption key and obtains the decrypted forensics data. In the meanwhile, $\mathcal{L}$ submits a transaction to prompt FSM into new state as: Data Collection $\xrightarrow{TX^6_{\mathcal{L}}}$ Data Examination.

$$TX^6_{\mathcal{L}}=[H_{req},H_0(H_{D_{id,type,t}}),Ti]_{K^s_{\mathcal{L}}}\tag{16}$$

4.2.5 Data Examination and Analysis

In this phase, $\mathcal{L}$ will comprehensively conduct systematic search of the collected data, which is to identify the potential evidences. Specifically, $\mathcal{L}$ needs to prove that the evidences belong to the legitimate collected data. Namely, the integrity of the evidences should be preserved. During the data examination, several kinds of software and hardware methods can be utilized as in [10]. Furthermore, data reduction can be performed to reduce the size of the outcome [42]. After the data examination, the state machine is transited to Data Analysis by $\mathcal{L}$ (denotes as $TX^7_{\mathcal{L}}$).

Data analysis is different from data examination that it is to further analyze the evidences collected in data examination phase. A number of techniques and tools can be used. We do not illustrate how to take advantage of these methods

to analyze the evidences, which is out of the scope of this paper. It is important to note that if $\mathcal{L}$ requires more data to be collected to accomplish the investigation, she/he needs to modify the court order and reapply for a request req^*, which will follow the typical flow to collect data. Otherwise, the state machine will be transited into Forensics Report (denotes as $TX^8_{\mathcal{L}}$).

4.2.6 Automated Vehicle Forensics Reporting

Based on the time-scheduling service in the designed contract, our scheme enables $\mathcal{L}$ to define notification on the final vehicle forensics report, so that the results and public information on this investigation is automatically sent to the different parties. For instance, if an accident happened and dispute existed, the final reports are required to send to drivers, insurance company, which can help them to make a judgement on the indemnity. The digest information of the report is recorded in the blockchain in case for need of the following investigation (denotes as $TX^9_{\mathcal{L}}$). If $\mathcal{C}$ and $\mathcal{L}$ jointly confirm the final report, the state machine will be transited into Complete (denotes as $TX^{10}_{\mathcal{C}}$, $TX^{10}_{\mathcal{L}}$).

According to the description of the above phases, we present the whole state machine transition for the vehicle digital forensics as the BB-VDF contract as shown in Fig. 5. In particular, when $\mathcal{L}$ accomplishes an investigation, the BB-VDF contract needs to receive a multiple signature [43] from $\mathcal{L}$ and $\mathcal{C}$ in the "Complete" phase to terminate the FSM instance.

BB-VDF Contract

On receive ("Request", des, H_{req}) from $\mathcal{L}$.
- create st;
- set st := WARRANT REQUEST;
- put warrrantPool[H_{req}] = (des, msg.sender, block.time);
- ▷ initialize the collected key shares number;
- initialize warrrantPool[H_{req}].sharesNum = 0;
- broadcast ("An investigation request has been created.");
- ▷ e.g., by Event mechanism on Ethereum

On receive ("Permit", H_{req}, $H_1(\rho(1))^{r_1}$, ..., $H_1(\rho(\ell))^{r_\ell}$) from $\mathcal{C}$.
- ▷ audit the warrant request off-chain.
- assert st = WARRANT REQUEST;
- parse req and check the validity as in Equation (12);
- put warrrantPool[H_{req}].keyPara = ($H_1(\rho(1))^{r_1}$, ..., $H_1(\rho(\ell))^{r_\ell}$);
- set st := WARRANT AUTHORIZATION;
- broadcast ("An investigation has been permitted by the court.");

On receive ("Reject", H_{req}) from $\mathcal{C}$.
- ▷ clean intermediate variables.
- delete warrrantPool[H_{req}];
- set st := COMPLETE;
- broadcast ("An investigation has been rejected by the court.");

On receive ("Store", H_{req}, $H_0(\varphi_1)$, ..., $H_0(\varphi_\ell)$) from $\mathcal{L}$.
- assert st = WARRANT AUTHORIZATION;
- put warrrantPool[H_{req}].keyBlindPara = ($H_0(\varphi_1)$, ..., $H_0(\varphi_\ell)$);
- broadcast ("The law enforcement submits the blinded values.");

On receive ("Retrieve", H_{req}, $g^{\mu_{1,j}} H_1(\rho(1))^{r_1 r'_j}$, ..., $g^{\mu_{\ell,j}} H_1(\rho(\ell))^{r_\ell r'_j}$) from A_j.
- assert st = WARRANT AUTHORIZATION;
- put warrrantPool[H_{req}].sharesPara = (π_1, $g^{\mu_{1,j}} H_1(\rho(1))^{r_1 r'_j}$, ..., $g^{\mu_{\ell,j}} H_1(\rho(\ell))^{r_\ell r'_j}$);
- warrrantPool[H_{req}].sharesNum += 1;
- ▷ check whether $t+1$ shares have been collected;
- **if** warrrantPool[H_{req}].sharesNum $\geq t+1$
  - set st := DATA COLLECTION;
  - broadcast ("The investigator collects enough shares.");
- **else** broadcast ("The A_j submits the decryption key shares.");

On receive ("Collect", H_{req}, $H_0(H_{M_{id,type,t_1}})$) from $\mathcal{L}$.
- assert warrrantPool[H_{req}].sharesNum $\geq t+1$;
- put warrrantPool[H_{req}].dataDigest=$H_0(H_{M_{id,type,t_1}})$;
- set st = DATA COLLECTION;
- broadcast ("The investigator has collected the forensics data.");

▷ NOTE: The process of data examination and analysis are skipped.

On receive ("Report", H_{req}, $H_0(report)$) from $\mathcal{L}$.
- assert st = DATA ANALYSIS;
- put warrrantPool[H_{req}].report=$H_0(report)$;
- set st := FORENSICS REPORT;
- broadcast ("The investigator starts to examine the forensics data.");

On receive ("Complete", H_{req}, $multiSignature$) from $\mathcal{L}$ and $\mathcal{C}$.
- verify multiSignature;
- assert st = FORENSICS REPORT;
- set st := COMPLETED;
- broadcast ("The investigation has been accomplished.");

Fig. 5. BB-VDF contract for the state machine.

5 Proof and Security Analysis

In this section, we present the proof of the correctness and discuss the security goals on accountability and privacy preservation of the proposed scheme.

5.1 Proof of Correctness

Lemma: If any each set $\mathcal{N}$ decryption authorities $\{\mathcal{A}_\tau\}_{\tau\in\mathcal{N}}$ provide the correct shares in the auditable data collection phase, i.e., $\{\widetilde{SK_\tau}\}_{\tau\in\mathcal{N}}$, $\mathcal{L}$ will surely recover the decryption key SK.

Proof: $\mathcal{L}$ receives $\widetilde{SK} = \{\widetilde{SK_\tau}\}_{\tau\in\mathcal{N}}$ from $t+1$ decryption authorities and uses $r_{\epsilon_1,d}, \epsilon_1 \in [1,\ell]$ to compute the following values:

$$\widetilde{SK_\tau} = \Big(\big(g^{\mu_{1,\tau}} H_1(\rho(1))^{r_1 r'_\tau}, g^{r_1 r'_\tau}, \forall d \in \Gamma/\rho(1), H_1(d)^{r_1 r'_\tau}\big), \ldots \big(g^{\mu_{\ell,\tau}} H_1(\rho(\ell))^{r_\ell r'_\tau}, g^{r_\ell r'_\tau}, \forall d \in \Gamma/\rho(\ell), H_1(d)^{r_\ell r'_\tau}\big)\Big), \tau \in \mathcal{N} \tag{17}$$

where $\mu_{a,\tau} = W_a \cdot \overrightarrow{v_\tau} = W_a \cdot (\alpha_\tau, y_2, ..., y_n), a \in [1,\ell]$. Then, $\mathcal{L}$ multiplies the same column values in $\{\widetilde{SK_\tau}\}_{\tau\in\mathcal{N}}$ together and uses the public λ_τ to multiple exponentially on the corresponding $\widetilde{SK_\tau}$. For instance, the product of the values in the a-th column of $\widetilde{SK}$ is denoted as follows:

$$\begin{aligned}
&\prod_{\tau\in\mathcal{N}} \Big((g^{\mu_{a,\tau}}) \cdot H_1(\rho(a))^{r_1 r'_\tau}\Big)^{\lambda_\tau} \\
&= g^{\sum_{\tau\in\mathcal{N}} \mu_{a,\tau}\lambda_\tau} \cdot H_1(\rho(a))^{r_1 \sum_{\tau\in\mathcal{N}} r'_\tau\cdot\lambda_\tau} \\
&= g^{\sum_{\tau\in\mathcal{N}} \overrightarrow{v_\tau}\cdot W_a\lambda_\tau} \cdot H_1(\rho(a))^{r_1 \sum_{\tau\in\mathcal{N}} r'_\tau\cdot\lambda_\tau} \\
&= g^{\sum_{\tau\in\mathcal{N}} (\alpha_\tau, y_2, ..., y_n)\cdot W_a\lambda_\tau} \cdot H_1(\rho(a))^{r_1 \sum_{\tau\in\mathcal{N}} r'_\tau\cdot\lambda_\tau} \\
&= g^{\sum_{\tau\in\mathcal{N}} (\lambda_\tau\cdot\alpha_\tau, \lambda_\tau y_2, ..., \lambda_\tau y_n)\cdot W_a} \cdot H_1(\rho(a))^{r_1 \sum_{\tau\in\mathcal{N}} r'_\tau\cdot\lambda_\tau} \\
&= g^{(\sum_{\tau\in\mathcal{N}} \lambda_\tau\cdot\alpha_\tau, \sum_{\tau\in\mathcal{N}} \lambda_\tau y_2, ..., \sum_{\tau\in\mathcal{N}} \lambda_\tau y_n)\cdot W_a} \cdot H_1(\rho(a))^{r_1 \sum_{\tau\in\mathcal{N}} r'_\tau\cdot\lambda_\tau} \\
&= g^{(\alpha, y'_2, ..., y'_n)\cdot W_a} \cdot H_1(\rho(1))^{r_1 \sum_{\tau\in\mathcal{N}} r'_\tau\cdot\lambda_\tau} \\
&= g^{\mu_a} \cdot H_1(\rho(a))^{r_1 \sum_{\tau\in\mathcal{N}} r'_\tau\cdot\lambda_\tau},
\end{aligned} \tag{18}$$

where $y'_l = \sum_{\tau \in \mathcal{N}} \lambda_\tau y_l, l \in [2, \ell]$.

Note that the other values of the exponent are kept the same, i.e., $r_1 \sum_{\tau \in \mathcal{N}} r'_\tau \cdot \lambda_\tau$. For example, the values in the 2-th column of SK is $g^{r_1 \sum_{\tau \in \mathcal{N}} r'_\tau \cdot \lambda_\tau}$. Thus, according to the KP-ABE scheme [30], $\mathcal{L}$ can recover the final decryption key SK.

5.2 Accountability

Complete Process Audit. As illustrated in Fig. 4, we model the process of vehicular digital forensics as a complete FSM, in which each state transition accomplishes with a blockchain transaction which contains a valid digital signature by the corresponding parties. The critical data records which are helpful for the auditability are stored in the transaction that no one can repudiate. Specially, the verification algorithms have been pre-defined in smart contracts. As long as the underlying $\mathcal{B}$ is secure, it is impossible for malicious attackers to disturb the normal running of smart contracts and prompt the state machine transition without being authorized and verified. Note that there are 8 states in our designed state machine and at least $(10 + t)$ transactions need be written into the blockchain (if $\mathcal{C}$ authorizes the warrant request), i.e., $TX^2_{\mathcal{L}}, TX^3_{\mathcal{C}}, TX^4_{\mathcal{L}}, \{TX^5_{\mathcal{A}_\tau}\}_{\tau \in \mathcal{N}}$ (t+1 transactions), $TX^6_{\mathcal{L}}, ..., Tx^{10}_{\mathcal{C}}, TX^{10}_{\mathcal{L}}$. These data records (similar to *audit logs* in [16]) on a specific warrant are recorded in the blockchain with tamper-resistance and traceability. Thus, BB-VDF has achieved complete process audit that malicious behaviors will be detected.

Public Verifiability. As analyzed above, the whole process of any investigation is recorded in $\mathcal{B}$. Although $\mathcal{B}$ is a permissioned-based blockchain in our design, while the data in $\mathcal{B}$ are not confidential that the public can read data in each transaction. Thus, the state on each warrant is transparent that any unauthorized or illegal behaviors will be detected by leveraging the smart contracts. In addition, as long as the defined security assumption holds, $\mathcal{L}$ can not acquire more data than the actually approved by $\mathcal{C}$. The public could verify whether the shares are computed based on the authorized warrant using the intermediate parameters. And also, zero knowledge proof can be utilized to publicly verify the validity of the shares-providing by $\mathcal{A}$.

Authorization. It is straightforward that authorization can be achieved by the *verification-then-forwarding* state machine. Before the data collection, it is necessary to submit $\mathcal{C}$'s digital signature for authorizing the warrant request to the BB-VDF smart contracts, which can ensure the legitimacy of the investigation. Furthermore, during the verification of a warrant, $\mathcal{C}$ will verify the validation of requested access policy $\mathbb{A} = (W, \rho)$, and generate the intermediate parameters (i.e., $\big(H_0(\rho(1))^{r_1}, ..., H_0(\rho(\ell))^{r_\ell}\big)$ which will be used to recover SK. During the collection of the secret shares for recovering the decryption key, each $\mathcal{A}_j$ will check whether the values are validated by $\mathcal{C}$ in $Tx^3_{\mathcal{C}}$. If no, $\mathcal{A}_j$ will refuse to provide the share.

5.3 Privacy Preservation

Confidentiality of Warrant. In our scheme, the secrecy of a warrant details is guaranteed against malicious or curious users. More preciously, during the process of warrant authorization, $\mathcal{L}$ sends the warrant details req to $\mathcal{C}$ through the secure channel without exposing to others. Besides, the attributions $\big(\rho(1), ..., \rho(\ell)\big)$ in $TX^3_{\mathcal{C}}$, $TX^4_{\mathcal{L}}$ and $\{TX^5_{\mathcal{A}_\tau}\}_{\tau \in \mathcal{N}}$ are blinded using the hash function $H_0(\cdot)$ and random numbers $(r_1, ..., r_\ell), \{r'_\tau\}_{\tau \in \mathcal{N}}$. Thus, even though the malicious users could enumerate the values of $H_0(\rho(x)), x = (\rho(1), ..., \rho(\ell))$, as long as $\mathcal{L}$ preserves these random numbers, malicious users can not learn about the attributions, which is based on the intractability of discrete logarithm problem (DLP).

Confidentiality of Forensics Data. The forensics data mainly refer to the data collected by the data sources $\mathcal{S}$. According to our design, $\mathcal{L}^*$ can not obtain the plaintext data directly from $\mathcal{S}$, but needs to be authorized by $\mathcal{C}$ to obtain the decryption key from multiple authorities $\mathcal{A}$. And also the public can audit the legitimacy of $\mathcal{L}^*$'s behaviors.

On the other hand, in terms of the honest-but-curious $\mathcal{C}$, there are two ways to obtain the secret key: first, 1) $\mathcal{C}$ colludes with $\mathcal{L}$ who can provide the decryption key, which is contrary to the security assumption. Second, 2) $\mathcal{C}$ colludes with the authorities $\mathcal{A} = \{\mathcal{A}_1, ..., \mathcal{A}_n\}$. However, we assume that at least more than t authorities are honest. Thus, according to the standard DKG protocol [29], it is impossible for $\mathcal{C}$ to restructure the decryption key SK with less than $t + 1$ shares. Herein, our scheme can resist against collusion attack between $\mathcal{C}$ and $\mathcal{A}$. Furthermore, as for the public values of $\{\widetilde{SK_\tau}\}_{\tau \in \mathcal{N}}$, since the random numbers $r_{\epsilon_1, d}$ are kept secrect, it is impossible for malicious users to restructure the final secret shares $\{SK_\tau\}_{\tau \in \mathcal{N}}$. As for $\mathcal{A}$, it is also apparent that they can not collude to recover the decryption key under the standard *t-of-n* threshold assumptions of secure DKG.

5.4 Data Security

Availability. Availability in our scheme means that the parties are able to conduct or participate in the vehicular digital forensics at anywhere and anytime. Put simply, it mainly includes two aspects: the available of data and the available of forensics service. First, the data is stored in distributed data storage system, which enables any party who has the decryption key and download link to download the data. Second, the underlying blockchain $\mathcal{B}$ is a decentralized architecture which can resist against distributed deny of service (DDoS) attacks, and exempt from SPoF/C. Thus, involved parties are able to access the blockchain-based system with high security to prompt the completion of an investigation on-chain.

Integrity. It is fairly straightforward that integrity can be guaranteed with the open blockchain. The digest information (i.e., the hash value of the forensics data) of data records are recorded in $\mathcal{B}$ using Merkle hash tree before an investigation, and no one can tamper these records. Thus, if $\mathcal{L}$ needs to present data evidences to the court, it is easy to prove the intergity of the data evidences using the Merkle verification algorithms. In addition, if the data are modified off chain, it can be easily detected with the hash values recorded in $\mathcal{B}$.

Unforgeability. Since each data record which will be used in the forensics and judgement should have the digital

signature, it is apparent for the unforgeability. Specifically, we consider two scenarios that malicious attackers could forge evidences to disrupt the normal forensics process in BB-VDF. First, $\mathcal{L}^*$ might use the expired warrant or create a forged warrant to acquire decryption key from the $\mathcal{A}$. However, the metadata of the warrant should be published in the BB-VDF contracts which will automatically check the timestamp on whether the warrant has expired. As long as no more than $t+1$ authorities provide correct shares, it is impossible for $\mathcal{L}^*$ to get the decryption key. Second, an unauthorized person may personate $\mathcal{L}$ to acquire data. However, due to the strict audit of $\mathcal{C}$, $\mathcal{A}$ and the permissioned blockchain nodes, the probability of the attack being successful is negligible.

6 Implementation and Evaluation Results

In this section, we present the implementation results of the proposed scheme which focus on three aspects: 1) communication and computation costs analysis; 2) experiments for the time performance analysis off-chain; 3) experiments for the transaction time performance analysis on-chain.

6.1 Simulation Design

We implement the cryptography algorithms on the DKG and KP-ABE protocol off-chain based on JPBC which is the PBC pairing-based cryptography library. H_0 is the JPBC built-in implementation of SHA256, and H_1 is the hash function that uses H_0 to convert the message to hash values and then maps these values to $\mathbb{G}_1$ (similar with H_2). We set the system parameters that based on the type A prime-order bilinear groups with 160_bit $\mathbb{Z}_p$ and 512_bit $\mathbb{G}_1$. Specially, we implement the cryptographic parts off-chain using *CloudCrypto* which is written in the Java program language[2]. CloudCrypto has supported the large universe KP-ABE protocol as in [44]. We modify the source code to enable it to support the time performance analysis of our proposed KP-ABE scheme.

We implement the experiments run on a personal computer ("Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz", 4GB RAM). We design DKG protocol based on 5 authorities (*3-of-5*) to evaluate the performance of the communication and computation costs. Following the standard KP-ABE protocol, we summarize the investigation process as four stage: Setup, KeyGen, Encryption and Decryption. The Setup phase includes the initialization of the DKG and KP-ABE protocols. Specially, The KeyGen phase includes the secret shares retrieval and secret key (i.e., SK) recovery. We count the time performance on Encryption and Decryption under the hybrid encryption method that mainly records the symmetric key encryption and decryption. The time cost on using symmetric encryption to encrypt and decrypt the original data are straightforward, which will not be analyzed in the experiments.

To show the practicability of the proposed scheme, we implement our state machine contract (BB-VDF contracts) in the official Ethereum public test network *Rinkeby* [3]. There have been already more than 3,305,000 accounts and 4,663,268 blocks in Rinkeby in 1 July, 2019. BB-VDF contracts are mainly composed of 14 sub-contracts. We evaluate the performance of our scheme by considering the network latency and transaction synchronous under the public blockchain network. We download *MetaMask* which allows us to connect to the chosen Rinkeby test network and run the smart contracts in browser without running a full blockcahin node [4]. The transaction fee is defined as the same for different transactions. We perform 20 sets of experiments in the Rinkeby. Each experiment creates a new life cycle of an FSM instance.

TABLE 2
Communication Cost in Setup, Encryption, Key Generation,and Decryption.

Phases	Communication Cost
Setup	$n^2(t \cdot \lfloor \mathbb{G}_1 \rfloor + 2\lfloor \mathbb{Z}_p \rfloor)$
Encryption	$\lfloor Enc(M) \rfloor + m\varsigma\lfloor H_0 \rfloor + \lfloor \mathbb{G}_1 \rfloor + \lfloor \mathbb{G}_T \rfloor$
KeyGen	$(n+t) \cdot (\ell^2 \lfloor H_1 \rfloor + \ell \lfloor \mathbb{G}_1 \rfloor) + \lfloor \mathbb{A} \rfloor + 2\ell t \lfloor H_1 \rfloor + \ell \lfloor H_0 \rfloor$
Decryption	$\lfloor Enc(M) \rfloor + \lfloor H_0 \rfloor$

[1] t is the number of authorities who submit the shares.
[2] The communication cost of PIR is not included.

6.2 Computation and Communication Cost Analysis

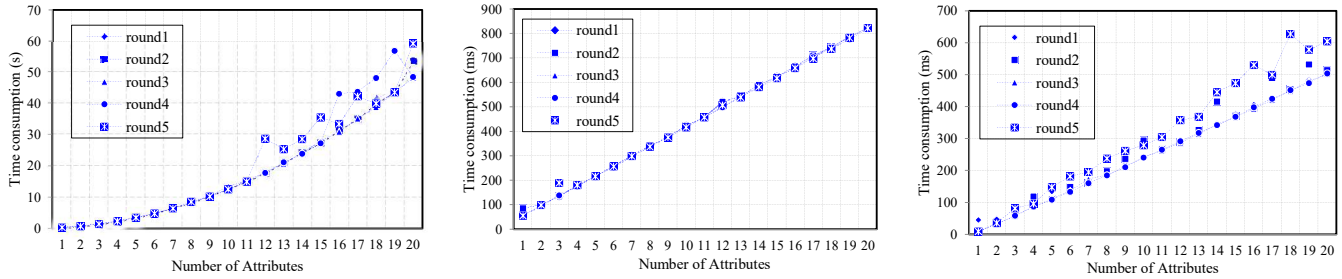
In this subsection, we discuss the computation and communication costs incurred by the proposed scheme. We mainly summarize the expensive computation and communication costs in the forensics process. Different notations are utilized to denote the operations in the corresponding group. Specifically, $\mathbb{G}_k$ denotes modular exponentiation in the corresponding group, i.e., $\mathbb{G}_1, \mathbb{G}_T$. H_k denotes the specific hash function, i.e., H_0, H_1 and H_2. $\mathbb{G}_k^m$ refers to the multiple modular exponentiation in group $\mathbb{G}_k$ with m bases. P denotes the bilinear pairing in KP-ABE. ℓ is the attributes number in the access policy $\mathbb{A}$. $\mathcal{N}$ denotes the number of authorities who participate in the secret share recovery. As for the vehicles, the main computation cost is mainly the data decryption, i.e., (Enc + $\mathbb{G}_T^2 + \ell \cdot \mathbb{G}_1 + G_1 + H_0 + 2H_2$). As for $\mathcal{L}$, the main computation cost lies in the data collection phase. Specially, after $\mathcal{C}$ returns res to $\mathcal{L}$, $\mathcal{L}$ blinds these values and computes the final decryption key, i.e., $(\ell+1) \cdot H_0 + \ell \cdot \big((\ell+1) \cdot \mathbb{G}_1 + \cdot 2\mathbb{G}_1 + (t+3) \cdot P + \mathbb{G}_1^{2\mathcal{N}} + 2 \cdot \mathcal{N} \cdot \mathbb{G}_1\big) + 2\mathbb{G}_1^{\mathcal{N}}$. As for $\mathcal{C}$, he authorizes a warrant and generates the initial parameters, i.e., $\ell \cdot \big((\ell+1) \cdot \mathbb{G}_1 + H_1\big)$. Besides, for a single authority, the main computation cost is to calculate $\widetilde{SK_j}$, i.e., $\ell \cdot \mathbb{G}_1^2 + \ell \cdot (\ell+1) \cdot \mathbb{G}_1$.

In terms of communication cost analysis, we use $\lfloor \mathbb{G}_x \rfloor$ to denote an element length in group $\mathbb{G}_x$. $\lfloor \mathbb{A} \rfloor$ refers to the size of the warrant. $\lfloor \mathbb{Z}_p \rfloor$ is used to refer to an element length in $\mathbb{Z}_p^*$. $\lfloor H_x \rfloor$ denotes the output length of different hash functions (e.g., H_0, H_1, H_2). Let $\lfloor M \rfloor$ be the size of the forensics data. The corresponding communication cost on different phases are shown in TABLE 2.

6.3 Off-chain Performance Evaluation

We evaluate the off-chain time performance for different phases. Specially, we conduct the experiments for 5 rounds. To test the impacts of attribute number on the computation

2. https://github.com/liuweiran900217/CloudCrypto
3. https://rinkeby.etherscan.io/
4. https://metamask.io/



(a) Key generation pahse (b) Encryption pahse (a) Decryption pahse

Fig. 6. Experimental results for the key generation, encryption and decryption algorithms with different attributes.

time, we use different number of attributes in each round. We only use "AND" in the access policy. According to our statistics, the average time consumption for the setup phase is about 35.91ms. Fig. 6 shows the time consumption versus the number of attributes. To be specific, the computation time costs for different phases are approximately linear with the increasing of the attributes. The key generation takes the most of the time compared with the other three phases. This is due to the fact that in order to recover SK, $\mathcal{C}$ needs to generate the parameters $\xi_1, ..., \xi_\ell$, and $\mathcal{L}$ computes the blinded values of these parameters. In addition, to recover the last secret key SK, $\mathcal{L}$ needs to compute the combined values with $\ell r_1 \sum_{\tau \in \mathcal{N}} r'_\tau \lambda_\tau$ exponentiations. In particular, we do not consider the network time delay among the different authorities. It is worth noting that as the attributes number is less than 4, the key generation is efficient that takes only 0.23s, 0.66s, 1.34s on average, respectively. As mentioned above, the size of the message used in encryption and decryption phase is small (i.e., the 128_bit symmetric key). As shown in Fig. 6 (b) and (c), the encryption and decryption phase are efficient with the increasing of attributes that take no more than 1s to encrypt and decrypt the message.

6.4 On-chain Performance Evaluation

In this subsection, we focus on the performance analysis of the state machine in the Rinkeby for a warrant. We record the time consumption for each transaction involved in the execution of the warrant. There are 10 type of transactions involved in the proposed scheme. As shown in Fig. 7, the average confirmation time for a specific transaction is bout 32.89s. Namely, each transaction takes about 2 blocks time to be finally confirmed in the Rinkeby (a new block is generated in every 15s). Note that the confirmation time is not related with the size of the transaction but the number of transactions in that specific time. Generally, the time consumption of transaction confirmation in the blockchain is acceptable. In order to achieve fast transaction confirmation, we can dynamically adjust the gas limitation for a block or decrease the consensus time in the customized blockchain network.

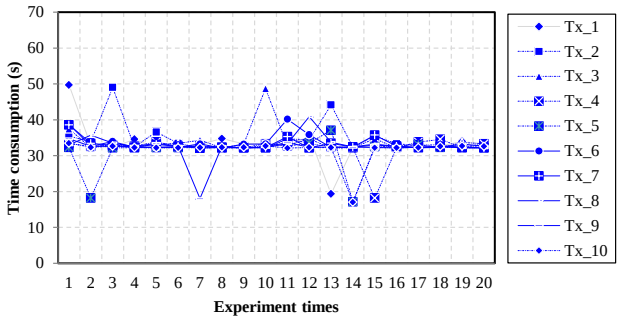


Fig. 7. The experiment results in reaching a consensus on verifying the validation of a transaction.

7 Related Work

7.1 Vehicular Digital Forensics

As for the vehicle forensics, the investigators need to master some forensics skills that include using the software and hardware tools, even dismantling and reassembling parts of the car [10], [17], [45]. There are two important components in the vehicle, i.e., insurance black box (IBB) and Event Data Recorder (EDR) [10]. Generally, IBB is a small "black box" device which records the car information, such as travelled distance, driving speed, braking, and cornering events. EDR is in charge of saving the data with regard to crash accident. Daily *et al.* proposed a developmental model for vehicle EDR forensics [45]. They utilized the digital forensics methods to preserve the information in EDR. Nhien-An *et al.* introduced some challenges and cases on VDF based on EDR in modern vehicles [10]. The discussed the hardware and software solutions for VDF. On the other hand, some works focused on how to allow authorized mobile devices to connect the car. Berla iVe[5] constructs a collection of software−hardware tools to support the investigators to conduct the vehicle forensics [17]. Mansor *et al.* proposed a vehicle forensics method that stored the data in remote cloud or server, which aimed to ensure the safety of the forensics data [17], [46].

7.2 Blockchain Technology for Digital Forensics

Many schemes on VDF have been proposed in recently years [10], [13], [14], [16], [17], [19], [35], [42]. Among them, Mumin *et.al* proposed a blockchain based framework for forensics applications of vehicles [14], which is the most relevant with our scheme. The proposed scheme connected different stakeholders to construct a permissoned blockchain network. Once an accident happened, their scheme could

5. https://berla.co/

reconstruct the accident scene and determine the faulty party. Compared with Mumin's scheme, we centered on two security goals that the process of the investigation should be accountable and preserve the privacy. The court and law enforcement are assumed to be honest-but-curious in BB-VDF, which is also the additional unsolved research problems in [14].

Decoster *et al.* designed HACIT2, a blockchain based application for dynamic navigation and forensics in VANET [47]. HACIT2 did not rely on the services of third parties while ensuring dynamic navigation rerouting and user anonymity. Auqib *et al.* proposed a blockchain-based digital forensics scheme named Forensic-Chain [18]. They aimed to ensure the integrity and tamper-resistant of data record by leveraging blockchain technology in digital forensics chain of custody, which was different with our security goals. In terms of the internet of things (IoT) forensics, Jung *et al.* [48] and Duc *et al.* [49] proposed a decentralized investigation framework for digital forensics. Their scheme aimed to ensure the integrity and non-repudiation of the IoT data, and enabled the investigation process transparent.

7.3 Accountability and Privacy in Digital Forensics

Researches on *accountability* and *privacy-preserving* in digital forensics have been worked for many years. Joshua and Dan have proposed a protocol for accountable warrant execution [16]. Their protocols guaranteed the accountability and secrecy of data records and requests. A secure IBE scheme with secret sharing of the master secret key is designed to resist SPoF/C. Jonathan *et al.* designed a practical accountable scheme for secret processes [13] to ensure the public could audit whether the surveillance powers were not abused. The authors considered that there were multiple courts who would exchange secret data and the data were stored in the company, which was different from the system model. Compared with [13] and [16], the the public ledger in [13] or the auditor in [16] were mainly utilized to maintain the data record with tamper-proof, while we leverage the blockchain technology to model the VDF process as a customized state machine in smart contracts.

8 Conclusion

In this paper, we analyzed the potential threats on security and privacy issues in VDF. We proposed BB-VDF on top of blockchain and smart contracts to enable accountability and fine-grained access control in this scenario. The whole vehicle forensics procedures are modeled as an FSM, which enables each party to cooperate honestly and auditably. We utilize the DKG protocol to manage the master secret key without relying on any third party. Based on this construction, the distributed KP-ABE with partially hidden access structures scheme is designed, which enables the fine-grained access control on forensics data. BB-VDF prevents malicious investigators to abuse or misuse powers and eliminates SPoF/C. The integrity of the forensics data retrieval is guaranteed with the underlying blockchain. Finally, we implemented the proposed scheme on the Ethereum public test network to test the feasibility and practicability.

Acknowledgments

This work was supported by National Key R&D Plan of China (Grant No. 2017YFB0802203, 2018YFB1003701), National Natural Science Foundation of China (Grant Nos. 61825203, U1736203, 61702222, 61472165, 61732021, 61877029, 61872153, 61802145, U1636209), National Joint Engineering Research Center of Network Security Detection and Protection Technology, Guangdong Provincial Special Funds for Applied Technology Research and Development and Transformation of Important Scientific and Technological Achieve (Grant Nos. 2016B010124009 and 2017B010124002), Guangdong Key Laboratory of Data Security and Privacy Preserving (Grant No. 2017B030301004), Guangzhou Key Laboratory of Data Security and Privacy Preserving (Grant No. 201705030004), China Postdoctoral Science Foundation (Grant No. 2017M612842).

References

[1] C. Huang, R. Lu, X. Lin, and X. Shen, "Secure automated valet parking: A privacy-preserving reservation scheme for autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11 169–11 180, 2018.

[2] "Automotive software systems complexity: Challenges and opportunities," "https://slideplayer.com/slide/15897985/", [Online].

[3] J. Wang, J. Liu, and N. Kato, "Networking and communications in autonomous driving: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1243–1274, 2018.

[4] A. V. Shvetsov, V. A. Sharov, and S. V. Shvetsova, "Method of protection of pedestrian zones against the terrorist attacks made by means of cars including off-road vehicles and trucks," *European Journal for Security Research*, vol. 2, no. 2, pp. 119–129, 2017.

[5] S. Perry, B. Hasisi, and G. Perry, "Who is the lone terrorist? a study of vehicle-borne attackers in israel and the west bank," *Studies in Conflict & Terrorism*, vol. 41, no. 11, pp. 899–913, 2018.

[6] "Nice attack death toll rises to 86 as injured man dies," "https://www.bbc.com/news/world-europe-37137816", [Online].

[7] "Rental cars have become terrorists choice," "https://www.businessinsider.com/home-depot-statement-after-new-york-city-truck-attack-2017-10", [Online].

[8] "Terrorist attacks by vehicle fast facts," "https://www.cnn.com/2017/05/03/world/terrorist-attacks-by-vehicle-fast-facts/index.html", [Online].

[9] J. Lacroix, K. El-Khatib, and R. Akalu, "Vehicular digital forensics: What does my vehicle know about me?" in *Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*. ACM, 2016, pp. 59–66.

[10] N.-A. Le-Khac, D. Jacobs, J. Nijhoff, K. Bertens, and K.-K. R. Choo, "Smart vehicle forensics: Challenges and case study," *Future Generation Computer Systems*, 2018.

[11] "Vehicle ramming," "https://www.dhs.gov/sites/default/files/publications/V Ramming-Security-Awareness-for-ST-CP", [Online].

[12] A. Nieto, R. Roman, and J. Lopez, "Digital witness: Safeguarding digital evidence by using secure architectures in personal devices," *IEEE Network*, vol. 30, no. 6, pp. 34–41, 2016.

[13] J. Frankle, S. Park, D. Shaar, S. Goldwasser, and D. Weitzner, "Practical accountability of secret processes," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 657–674.

[14] M. Cebe, E. Erdin, K. Akkaya, H. Aksu, and S. Uluagac, "Block4forensic: An integrated lightweight blockchain framework for forensics applications of connected vehicles," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 50–57, 2018.

[15] S. Zawoad, A. K. Dutta, and R. Hasan, "Towards building forensics enabled cloud through secure logging-as-a-service," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 2, pp. 148–162, 2015.

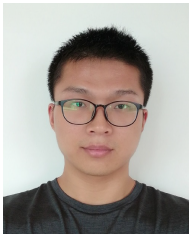
[16] J. Kroll, E. Felten, and D. Boneh, "Secure protocols for accountable warrant execution," *See http://www. cs. princeton. edu/felten/warrant-paper. pdf*, 2014.

[17] H. Mansor, K. Markantonakis, R. N. Akram, K. Mayes, and I. Gurulian, "Log your car: The non-invasive vehicle forensics," in *2016 IEEE Trustcom/BigDataSE/ISPA*. IEEE, 2016, pp. 974–982.

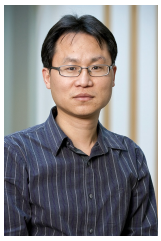
[18] A. H. Lone and R. N. Mir, "Forensic-chain: Blockchain based digital forensics chain of custody with poc in hyperledger composer," *Digital Investigation*, vol. 28, pp. 44–55, 2019.
[19] U. Karabiyik, "Building an intelligent assistant for digital forensics," 2015.
[20] K. Fanning and D. P. Centers, "Blockchain and its coming impact on financial services," *Journal of Corporate Accounting & Finance*, vol. 27, no. 5, pp. 53–57, 2016.
[21] T. V. Asaph Azaria, Ariel Ekblaw, "Medrec: Using blockchain for medical data access and permission management," in *2nd International Conference on Open and Big Data, OBD 2016*, Vienna, Austria, Aug. 2016, pp. 25–30.
[22] D. Liu, A. Alahmadi, J. Ni, X. Lin *et al.*, "Anonymous reputation system for iiot-enabled retail marketing atop pos blockchain," *IEEE Transactions on Industrial Informatics*, 2019.
[23] R. S. M. J. F. Muneeb Ali, Jude Nelson, "Blockstack: A global naming and storage system secured by blockchains," in *USENIX Annual Technical Conference, USENIX ATC 2016*, Denver, CO, 2016, pp. 181–194.
[24] M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J.-N. Liu, Y. Xiang, and R. Deng, "Crowdbc: A blockchain-based decentralized framework for crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, 2018.
[25] Y. Lu, Q. Tang, and G. Wang, "Zebralancer: Private and anonymous crowdsourcing system atop open blockchain," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018, pp. 853–865.
[26] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009.
[27] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.
[28] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Annual International Cryptology Conference*. Springer, 1991, pp. 129–140.
[29] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1999, pp. 295–310.
[30] S. Hohenberger and B. Waters, "Attribute-based encryption with fast decryption," in *International Workshop on Public Key Cryptography*. Springer, 2013, pp. 162–179.
[31] Z. Liu, Z. Cao, and D. S. Wong, "Efficient generation of linear secret sharing scheme matrices from threshold access trees," *Cryptology ePrint Archive: Listing*, 2010.
[32] N. H. Ab Rahman, W. B. Glisson, Y. Yang, and K.-K. R. Choo, "Forensic-by-design framework for cyber-physical cloud systems," *IEEE Cloud Computing*, vol. 3, no. 1, pp. 50–59, 2016.
[33] A. Yang, J. Weng, N. Cheng, J. Ni, X. Lin, and X. Shen, "Deqos attack: Degrading quality of service in vanets and its mitigation," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4834–4845, May 2019.
[34] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Annual International Cryptology Conference*. Springer, 2017, pp. 357–388.
[35] O. Henniger, "Evita: E-safety vehicle intrusion protected applications," *tech. rep., EVITA*, 2011.
[36] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2015, pp. 281–310.
[37] H. Duan, Y. Zheng, Y. Du, A. Zhou, C. Wang, and M. H. Au, "Aggregating crowd wisdom via blockchain: A private, correct, and robust realization."
[38] "Ethereum alarm clock," "https://www.ethereum-alarm-clock.com/", [Online].
[39] S. Dziembowski, L. Eckey, and S. Faust, "Fairswap: How to fairly exchange digital goods," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 967–984.
[40] H. Yang, W. Shin, and J. Lee, "Private information retrieval for secure distributed storage systems," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 12, pp. 2953–2964, 2018.
[41] C. Rackoff and D. R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," in *Annual International Cryptology Conference*. Springer, 1991, pp. 433–444.
[42] S. Ballou, *Electronic crime scene investigation: A guide for first responders*. Diane Publishing, 2010.
[43] R. O'Connor, "Simplicity: A new language for blockchains," in *Proceedings of the 2017 Workshop on Programming Languages and Analysis for Security*. ACM, 2017, pp. 107–120.
[44] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 463–474.
[45] J. S. Daily, N. Singleton, B. Downing, and G. W. Manes, "Light vehicle event data recorder forensics," in *Advances in Computer and Information Sciences and Engineering*. Springer, 2008, pp. 172–177.
[46] A. Yang, J. Xu, J. Weng, J. Zhou, and D. S. Wong, "Lightweight and privacy-preserving delegatable proofs of storage with data dynamics in cloud storage," *IEEE Transactions on Cloud Computing*, 2018.
[47] D. Kevin and B. David, "Hacit2: A privacy preserving, region based and blockchain application for dynamic navigation and forensics in vanet," in *International Conference on Ad Hoc Networks*. Springer, 2018, pp. 225–236.
[48] J. H. Ryu, P. K. Sharma, J. H. Jo, and J. H. Park, "A blockchain-based decentralized efficient investigation framework for iot digital forensics," *The Journal of Supercomputing*, pp. 1–16, 2019.
[49] D.-P. Le, H. Meng, L. Su, S. L. Yeo, and V. Thing, "Biff: A blockchain-based iot forensics framework with identity privacy," in *TENCON 2018-2018 IEEE Region 10 Conference*. IEEE, 2018, pp. 2372–2377.

Ming Li received his B.S. in electronic information engineering from University of South China in 2009, and M.S. in information processing from Northwestern Polytechnical University in 2012. From 2016, he started his Ph.D. at Jinan University. His research interests include blockchain, crowdsourcing, and its privacy and security.

Jian Weng is a professor and the Executive Dean with College of Information Science and Technology in Jinan University. He received B.S. degree and M.S. degree from South China University of Technology in 2001 and 2004 respectively, and Ph.D. degree at Shanghai Jiao Tong University in 2008. His research areas include public key cryptography, cloud security, blockchain, etc. He has published 80 papers in international conferences and journals such as CRYPTO, EUROCRYPT, ASIACRYPT, TCC, PKC, CT-RSA, IEEE TPAMI, IEEE TDSC, etc. He received the Young Scientists Fund of the National Natural Science Foundation of China in 2018, and the Cryptography Innovation Award from Chinese Association for Cryptologic Research (CACR) in 2015. He served as General Co-Chair for SecureComm 2016, TPC Co-Chairs for RFIDsec'13 Asia and ISPEC 2011, and program committee members for more than 40 international cryptography and information security conferences. He also serves as an associate editor of IEEE Transactions on Vehicular Technology.



Jia-Nan Liu is a Ph.D. student of Jinan University. He was born in July, 1992. He received B.S. degree and M.S. degree at Zhengzhou University and Jinan University in 2013 and 2016 respectively. His research interesting includes cryptography and cloud computing security.



Xiaodong Lin (M'09-SM'12-F'17) received the PhD degree in Information Engineering from Beijing University of Posts and Telecommunications, China, and the PhD degree (with Outstanding Achievement in Graduate Studies Award) in Electrical and Computer Engineering from the University of Waterloo, Canada. He is currently an associate professor in the School of Computer Science at the University of Guelph, Canada. His research interests include computer and network security, privacy protection, applied cryptography, computer forensics, and software security. He is a Fellow of the IEEE.

Charlie Obimbo is an associate professor in the School of Computer Science at the University of Guelph, Canada. He received his Ph.D. in 2000 from the University of New Brunswick in Canada. He has worked as an assistant professor both at the University of New Brunswick, and the University of Prince Edward Island, before joining the University of Guelph in 2001. His areas of research include computer and network security, applied cryptography, and analysis and design of computer algorithms.