Secure Data Sharing With Flexible Cross-Domain Authorization in Autonomous Vehicle Systems

Jianfei Sun, Guowen Xu, Tianwei Zhang, *Member, IEEE*, Xiaochun Cheng, Xingshuo Han, and Mingjian Tang

***Abstract*— As an increasingly prevalent technology in intelligent autonomous transportation systems, autonomous vehicle platoon has been indicated the ability to significantly reduce fuel consumption as well as heighten highway safety and throughput. However, existing efforts rarely focus on protecting data confidentiality and authenticity in autonomous vehicle platoons. How to ensure secure and high-fidelity platoon-level communication is still in its infancy. This paper makes the first attempt for efficient and secure communication across autonomous vehicle platoons. Specifically, we present `PDSM-FC`, the first privacy-preserving data share mechanism with flexible cross-domain authorization over distinctive platoons. The key insight of `PDSM-FC` is the design of a new ciphertext conversion technique, which allows a ciphertext to be easily converted into another type of ciphertext, facilitating efficient access by all entities holding the legitimate authorization. As a result, `PDSM-FC` can achieve high-fidelity data communication between two unique platoons in ciphertext, so as to complete specific tasks including platoon integration. Rigorous security analysis shows that `PDSM-FC` is secure against various attacks such as collusion, forgery and chosen-plaintext attacks. Moreover, theoretical evaluation and extensive experiments demonstrate the practicability of `PDSM-FC` in terms of functionality, storage and computation overheads.**

***Index Terms*— Autonomous vehicle, authenticity, high-fidelity, cross-domain authorization.**

I. Introduction

THE fast advancement of 5G, wireless communication, sensing, and artificial intelligence technologies has significantly accelerated the transition from traditional manual vehicle systems to intelligent autonomous transportation systems [1]–[3]. Many industrial giants have released mature autonomous driving products to our daily life, such as Tesla Autopilot system [4], Google Waymo vehicle [5], and Baidu Apollo platform [6]. As one of the most promising autonomous vehicle technologies, vehicle platooning not only realizes groups of close-following vehicles (called platoon or convoy) driving together with low headway, but also facilitates information sharing and dissemination among autonomous vehicles. A vehicle platooning scenario generally includes two common tasks: (1) maintaining a reasonable shape of a platoon. This starts with the platoon leader (PL) sending to its platoon followers (PFs) a series of data, including around-turning, braking, and deceleration, and then PFs follow the instructions to maintain the desired formation. (2) Consolidating two platoons to reduce road redundancy, thereby increasing the traffic throughput [7]–[9]. To achieve this, the PL of one platoon is required to share its platoon information (e.g., platoon identity, platoon size, platoon member) to the PL of another platoon, which then tells all its PFs to change the platoon leader as the PL of the first platoon.

Despite the numerous benefits the platooning brings, it has also raised widespread concern regarding privacy and security threats [10]–[12]. This can be summarized from two perspectives. First, current platooning systems lack the protection of data confidentiality and authenticity. For confidentiality, the interactive information between vehicles, including location and speed, is often highly sensitive and should only be accessed by authorized vehicles. Existing data transmission in plaintext inevitably incurs diversified privacy threats. For authenticity, each vehicle performs concrete operations according to the received instructions (e.g., turning, decelerating, turning around). Since there is no preset data authenticity verification strategy, a compromised vehicle is fully capable of tampering with or forging legal instructions to destroy the integrity of the mission.

To achieve the protection of data confidentiality and authenticity simultaneously, one possible solution is to combine the state-of-the-art digital signature techniques with the traditional encryption mechanisms [13]–[15], which empowers users to sign-then-encrypt their data. However, this solution is generally infeasible since heavy computational overheads are involved and the security cannot be exactly guaranteed as the respective technologies. An alternative solution is to utilize conventional signcryption technologies [16]–[18], which allow users to encrypt and sign their data together, such that only authorized users can validate their authenticity and access them. However, most existing signcryption schemes are computationally inefficient due to the heavy pairing computation operations involved. Hence, how to efficiently guarantee the authenticity and confidentiality of data is a prime challenge.

Manuscript received 13 December 2021; revised 9 February 2022; accepted 28 February 2022. Date of publication 16 March 2022; date of current version 7 July 2023. This work was supported in part by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 under Grant RG108/19 (S) and in part by the Nanyang Technological University (NTU)-DESAY SV Research Program under Grant 2018-0980. The Associate Editor for this article was S. Mumtaz. *(Corresponding author: Guowen Xu.)*

Jianfei Sun, Guowen Xu, Tianwei Zhang, and Xingshuo Han are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: jianfei.sun@ntu.edu.sg; guowen.xu@ntu.edu.sg; tianwei.zhang@ntu.edu.sg; xingshuo001@e.ntu.edu.sg).

Xiaochun Cheng is with the Department of Computer Science, Middlesex University London, London NW4 4BT, U.K. (e-mail: x.cheng@mdx.ac.uk).

Mingjian Tang is with Westpac Banking Corporation, Sydney, NSW 2000, Australia (e-mail: tmj2000@hotmail.com).

Digital Object Identifier 10.1109/TITS.2022.3157309

1558-0016 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on July 10,2023 at 04:23:22 UTC from IEEE Xplore. Restrictions apply.

Second, it is challenging to realize efficient data communication between different platoons in ciphertext. Considering a scenario where there are two platoons (A and B) traveling on the road with the same direction. At certain moment, these two platoons are required to merge into one platoon for ease of the task implementation needs. We assume that platoon B will be merged into A, and the PL of the merged platoon is the PL of platoon A (i.e., PL^A). To successfully merge platoon B, PL^A needs to share some platoon information (e.g., platoon id, platoon size, platoon member, platoon location, etc.) with the PL of platoon B (i.e., PL^B), which then tells all its PFs to switch the platoon leader to PL^A. To preserve the platoon information privacy, PL^A commonly encrypts the platoon information by some encryption mechanism, e.g., identity-based encryption (IBE), so that only PL^B can read the related information. To enable sharing platoon A's information with the PFs of platoon B in real-time, possible solutions are mainly originated from the following four strategies.

A. Decryption- Then-Broadcasting

PL^B first decrypts all the encrypted platoon information and then sends it to its PFs in plaintext. Obviously, this is impractical for PL^B since the frequent decryption-then-broadcasting operations can clearly result in considerable computation and communication costs. Furthermore, the data transmission in plaintext can cause privacy leakage risk of the platoon information. Besides, PL^B may first decrypt all encrypted platoon information and then conduct multiple encryption operations under different public keys of various platoon vehicles to realize one-to-many platoon data sharing. It is clearly inefficient since the computation and communication costs are linearly growing with the increment of the number of shared platoon vehicles.

B. Broadcasting With Attached Secret Keys

PL^B who can access the original platoon data encrypts its private key to be shared with all its PFs, such that each of the authorized vehicle can obtain the private key, thus accessing the original data. Clearly, this solution is also inadvisable since the direct exposure of the authorized vehicle's private key would result in decryption privilege abuse issues.

C. Proxy Re-Encryption Technology [20]–[25]

PL^B implements the conversion of ciphertext of platoon A for different platoon vehicles in the same public key setting, such that the converted data can be only accessible by authorized vehicles in the same system. However, this technology can only transform the ciphertext in one format to another one in the same ciphertext format [26]–[32], which makes the vehicles of platoon B in different systems inaccessible to the ciphertext of platoon A.

D. Cross-Domain Transformation Technology [33]–[35]

This technique enables ciphertext transformation between various platoon vehicles under the distinct public key settings. Specifically, it allows PL^B to transform the data of platoon A in one ciphertext format to that in another ciphertext format, such that the vehicles in platoon B can succeed in accessing the data of platoon A. However, the existing cross-domain encryption approaches suffer from either serious system efficiency problems due to the need for secure interaction between various entities, or vulnerabilities against some malicious attacks, leading to data privacy risks.

In summary, how to find an efficient solution to achieve practical and secure data sharing between two various platoons is unsolved. Driven by the above challenges, this paper designs the first-ever Privacy-preserving Data Share Mechanism with Flexible Cross-domain authorization (PDSM-FC) in intelligent autonomous vehicle platoons. PDSM-FC elegantly incorporates the privacy-preserving matchmaking encryption technology [37] into the cross-domain transformation primitive [36] to simultaneously realize the authenticity of platoon data and the cross-platoon data sharing. The main contributions of our paper are summarized as below:

- **Authenticity of shared platoon data**. To ensure platoon A's data to be authentically shared with PL and PFs of platoon B, the proposed scheme empowers PL^A to embed a signature related to its identity into the encrypted platoon data, such that the platoon data get rid of the risks of being forged, replaced and tampered with.
- **Flexible cross-domain authorization**. To realize cross-domain authorization for multiple platoon vehicles via PL^B, the proposed scheme permits PL^B to transform the data in the format of IBE ciphertext to that in the IBBE cipheretext format, such that multiple PFs of platoon B have the same privilege as PL^B to read the encrypted platoon data.
- **Strong security assurance**. To guarantee strong security, the proposed scheme cannot only block any unauthorized access to the platoon data stored in the cloud but also hamper the disclosure of any useful information about the platoon data during the whole cross-domain transformation.

We give detailed security proofs and analysis to indicate that our proposed PDSM-FC is selectively secure, decryption key leakage-resistant, and collusion resistant in the oracle model. Further, we perform extensive experimental evaluations and comparisons with some related works, to demonstrate the high practicality and superiority of our approach in transforming the encrypted platoon data and authorization access.

II. Related Works

A. Inter/Cross-Domain Data Transformation

The primitive of proxy re-encryption (PRE) was put forward by Blaze *et al.* [20], which is used for handling inter-domain data transformation within the same cryptographic encryption system. Specifically, PRE enables transforming a ciphertext produced under Alice's public key into a ciphertext under Bob's public key in the same public key setting. It has been classified into several categories: unidirectional and bidirectional PRE, non-interactive and interactive PRE, single and multi-hop PRE. Over the years, a large number of PRE research works were conducted, to continuously improve the functionality, efficiency and security. For example, the first unidirectional PRE was formally formulated by Libert and

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on July 10,2023 at 04:23:22 UTC from IEEE Xplore. Restrictions apply.

Vergnaud [21] formally formulated the first unidirectional PRE. Then, Guo *et al.* [22] introduced the accountability into the unidirectional PRE scheme to discern the malicious proxy server that abuses the re-encryption keys. To further enrich the functionality of PRE, the first identity-based PRE (IBPRE) was designed via a technical combination of identity-based encryption (IBE) and PRE by Green and Ateniese [23], which is commonly considered as an extension of PRE based on the identity-based setting. To resolve the inefficiency problem, an IBPRE scheme with constant decryption keys and ciphertext was invented by Chu and Tzeng [24], whereas it easily suffers from collusion attacks that lead to the confidentiality-loss of encrypted data. To combat this security defect, Liang *et al.* [25] suggested a cloud-based IBPRE scheme with revocability. However, for each data transformation, an interaction between the data owner and the key generation authority is needed, which practically results in an inefficiency issue. Xu *et al.* [26] proposed a conditional identity-based broadcast proxy re-encryption scheme (IBBE-PRE) by introducing identity-based broadcast encryption (IBBE) into PRE: a message attached to the specified receivers' identities is encrypted and sent to multiple receivers to decipher and a re-encryption key can be produced and delegated to a proxy, which transforms the original ciphertext into a novel one that can be decrypted by a new set of target receivers. Besides, PRE was also extended to construct attribute-based PRE (ABPRE) [27], [28], function-based PRE [29], [30] and homomorphic PRE [31], [32]. However, these PRE constructions mainly enable the ciphertext conversion in the same cryptographic encryption proposals instead of realizing the ciphertext transformation from one format to another format.

To provide cross-domain ciphertext transformation, Matsuo [33] proposed a solution to transform a ciphertext of a public key encryption (PKE) system into that of an IBE system by incorporating PRE into the linked IBE and PKE. However, this solution fails to realize the sharing of converted data with multiple users. Besides, Mizuno and Doi [34] introduced a unidirectional PRE approach that realizes the transformation of a ciphertext of an attribute-based encryption system into that of an IBE system. Nevertheless, it requires users to securely communicate with each other while storing some necessary information to conduct the transformation. Jiang *et al.* [35] invented a cross-domain ciphertext transformation scheme between a conventional PKE and IBE, in which each participant requires certificates to authenticate the legitimacy of each other. Until now, there is no such a scheme that can realize a ciphertext transformation from a single-receiever IBE system to a multi-recipient IBBE system while achieving data authenticity. In this work, we attempt to invent an encryption transformation scheme with data forgery-resistance, which achieves data authenticity and flexible cross-platoon authorization with strong security guarantee for the vehicle platoon applications.

B. Privacy-Preserving Vehicle Communications

A variety of privacy-preserving solutions have been proposed to protect the vehicle networking systems, e.g., vehicular ad hoc networks (VANETs) and vehicle social networks (VSN) [39]–[41]. Specifically, these works mainly focus on three scenarios: (1) cross-vehicle authentication [42], [43]; (2) vehicle-to-vehicle communication [44], [45]; (3) vehicle platoon splitting [46], [47].

Those works mainly consider the vehicle communication within the same platoon. Hence, each vehicle participant holds the same public parameters in a cryptographic system. Different from these works, we focus on the vehicle platoon merging scenario, which involves communication across different platoons. Vehicles in each platoon have their own independent cryptographic public parameters, even if the same cryptographic primitive is used. This brings a couple of new challenges to design privacy-preserving solutions. (1) Existing solutions cannot be directly applied to the cross-platoon vehicle-to-vehicle communication in vehicle platoon merging since the distinct communication protocols are possibly adopted in different vehicle platoons. (2) The cross-domain transformation technology enables secure vehicle-to-vehicle communication between various platoon vehicles under the distinct public key settings. However, existing cross-domain encryption approaches suffer from serious efficiency problems due to the need for secure interactions between various entities. Motivated by these challenges, the goal of this paper is to design a practical solution customized for the specific cross-platoon communication scenario with high security and efficiency while preserving data confidentiality and authenticity during the whole communication. To this end, we adopt the privacy-preserving matchmaking encryption technology and the cross-domain transformation primitive for simultaneously realizing the platoon data authenticity and the cross-platoon vehicle-to-vehicle communication.

TABLE I
NOTATIONS

Notation	Description
pp	Public parameter
sk	Secret key
ek	Encryption key
at	Authorized token
$\mathsf{ct}_{\mathsf{id}}$	Ciphertext for the vehicle with its identity id
$\mathsf{ct}_{\mathcal{S}}$	Ciphertext for the set $\mathcal{S}$ of vehicles

III. Preliminaries and Problem Statement

In this section, we first introduce the basic knowledge for our proposed methodology, including bilinear maps and complexity assumptions. We also present the problem statement involving the system architecture, threat model. Table I shows the notations used in our methodology.

A. Bilinear Maps and Complexity Assumptions

Definition 1 (Bilinear Maps): Consider two multiplicative cyclic groups $\mathbb{G}_0$ and $\mathbb{G}_1$ with the same order p, where g is a random generator of group $\mathbb{G}_0$. The bilinear map relationship

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on July 10,2023 at 04:23:22 UTC from IEEE Xplore. Restrictions apply.

for $\mathbb{G}_0$ and $\mathbb{G}_1$ is defined as $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$, which has the following properties: (1) $e(v^x, h^y) = e(v, h)^{xy}$, where $v, h \in \mathbb{G}_0$, $x, y \in \mathbb{Z}_p$. (2) $e(g, g) \neq 1$.

Definition 2 (Bilinear Diffie-Hellman (BDH) Assumption): Given a BDH instance (P, P^x, P^y, P^z), where P is a random element of $\mathbb{G}_0$, it is intractable to compute $\mathcal{Z} = e(P, P)^{xyz}$.

Definition 3 (Sequence Diffie-Hellman (SDH) Assumption): Given an ℓ-SDH instance $(g_0, g_0^{\theta}, g_0^{\theta^2}, \ldots, g_0^{\theta^{\ell}}, \rho)$, it is difficult to get the result $g_0^{1/(\theta+\rho)}$.

Definition 4 (Generic Decisional Diffie-Hellman Exponentiation (GDDHE) Assumption): Given a GDDHE instance $(g_0^{\theta}, \ldots, g_0^{\theta^{\ell-1}}, g_0^{\theta\mathcal{P}(\theta)}, g_0^{r_1\theta\mathcal{P}(\theta)}, u_0, u_0^{\theta}, \ldots, u_0^{\theta^{2k}}, u_0^{r_1\theta\mathcal{Q}(\theta)}, g_0, \mathcal{Z})$ of GDDHE assumption, it is hard to decide $\mathcal{Z} = e(g_0, u_0)^{r_1\mathcal{P}(\theta)}$, or a random element of $\mathbb{G}_1$, where two different generators g_0, u_0 of cyclic group $\mathbb{G}_0$ are set and two polynomials $\mathcal{P}(x) = \prod_{i=1}^{\ell}(x + \rho_i)$, $\mathcal{Q}(x) = \prod_{i=\ell+1}^{\ell+k}(x + \rho_i)$ own various roots.

B. Outline of PDSM-FC Scheme

Our PDSM-FC is constructed by the following algorithms:

- Setup (1^{λ}, ℓ): Input a security parameter 1^{λ} and the allowed maximum number ℓ of users to simultaneously access some data, output a system public parameter pp and a system master secret key msk.
- EKGen (id*, msk): Input pp, msk and an identity id*, output an encryption key $\mathsf{ek}_{\mathsf{id}^*}$.
- SKGen (pp, msk, id): Input pp, msk and an identity id, output a secret key $\mathsf{sk}_{\mathsf{id}}$.
- Encrypt (pp, M, id, $\mathsf{ek}_{\mathsf{id}^*}$): Input pp, $\mathsf{ek}_{\mathsf{id}^*}$, the message M and an identity id, output an IBE ciphertext $\mathsf{ct}_{\mathsf{id}}$.
- Authorize (pp, $\mathsf{sk}_{\mathsf{id}}$, $\mathcal{S}$): Input pp, $\mathsf{sk}_{\mathsf{id}}$ and the identity set $\mathcal{S}$ of data receivers, output an authorized token $at_{\mathsf{id}\rightarrow\mathcal{S}}$.
- Transform (pp, $at_{\mathsf{id}\rightarrow\mathcal{S}}$, $\mathsf{ct}_{\mathsf{id}}$): Input pp, $at_{\mathsf{id}\rightarrow\mathcal{S}}$ and $\mathsf{ct}_{\mathsf{id}}$, output a new transformed IBBE ciphertexts $\mathsf{ct}_{\mathcal{S}}$.
- Decrypt (pp, $\mathsf{ct}_{\mathsf{id}}/\mathsf{ct}_{\mathcal{S}}$, $\mathsf{sk}_{\mathsf{id}'}$): Input pp, $\mathsf{ct}_{\mathsf{id}}/\mathsf{ct}_{\mathcal{S}}$ and $\mathsf{sk}_{\mathsf{id}'}$, output M if $\mathsf{id} = \mathsf{id}'$ or $\mathsf{id}' \in \mathcal{S}$, and $\perp$ otherwise.

Correctness: For an IBE ciphertext, $\mathsf{ct}_{\mathsf{id}} \leftarrow$ Encrypt(pp, M, id, $\mathsf{ek}_{\mathsf{id}^*}$) and $\mathsf{sk}_{\mathsf{id}'} \leftarrow$ SKGen(pp, msk, id′). If $\mathsf{id} = \mathsf{id}'$, the decryption algorithm can recover $M \leftarrow$ Decrypt(pp, $\mathsf{ct}_{\mathsf{id}}$, $\mathsf{sk}_{\mathsf{id}'}$). For an IBBE ciphertext, $\mathsf{ct}_{\mathcal{S}} \leftarrow$ Transform(pp, $at_{\mathsf{id}\rightarrow\mathcal{S}}$, $\mathsf{ct}_{\mathsf{id}}$), where $\mathsf{ct}_{\mathsf{id}} \leftarrow$ Encrypt(pp, M, id, $\mathsf{ek}_{\mathsf{id}^*}$), $\mathsf{at}_{\mathsf{id}\rightarrow\mathcal{S}} \leftarrow$ Authorize(pp, $\mathsf{sk}_{\mathsf{id}}$, $\mathcal{S}$) and any secret key $\mathsf{sk}_{\mathsf{id}'} \leftarrow$ SKGen(pp, msk, id′). If $\mathsf{id}' \in \mathcal{S}$, the decryption algorithm can recover $M \leftarrow$ Decrypt(pp, $\mathsf{ct}_{\mathcal{S}}$, $\mathsf{sk}_{\mathsf{id}'}$).

C. System Architecture

Fig. 1 shows the architecture of vehicle platoon management with our PDSM-FC, which involves four types of participants: data owners, data users, vehicle cloud server unit (VSU), registry authority (RA). Data owners and users are vehicle platoon clients. RA (such as IT center of vehicle authority) is a fully reliable entity that takes charge of initializing the system, generating the public parameters for the other system entities, and issuing secret keys for clients to complete user registration. The data owner is the platoon leader who is responsible for sharing the platoon information, such as platoon location, identity, size, etc., with platoon users of other vehicle platoons. Data users are vehicle platoon users who are responsible for receiving and performing the order of platoon management via the shared data from data owners. VSU has two major capabilities: (i) the storage service is rendered for clients to store platoon information. (ii) the computation service is accommodated for clients to transform the stored vehicle platoon data in one ciphertext format into that in another ciphertext format.

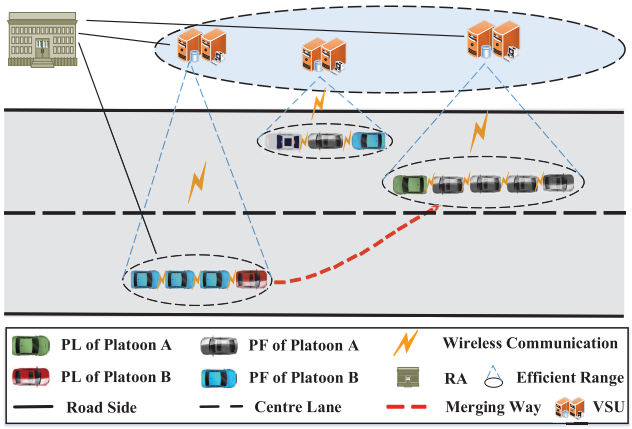


Fig. 1. System architecture of vehicle platoon management.

To successfully merge a new platoon, data owner (PL^A) outsources its platoon information with a data user (PL^B), which subsequently transfers the platoon A's information to its PFs for performing the merging task. Specifically, to preserve the privacy of its platoon information, PL^A encrypts its data with an IBE scheme and then outsources the encrypted data to VSU for sharing with PL^B for ease of platoon merging. To better perform merging of these two platoons, PL^B requires to share the data with its PFs. In this way, PL^B creates an authorization token and sends it to VSU for transforming the IBE ciphertext format into an IBBE ciphertext format, such that the PFs can recover and capture the data of PL^A. In this way, the IBE ciphertext of PL^A formerly accessible to only PL^B can be authorized to the PFs of platoon B for access.

D. Threat Model and Security Requirements

In our PDSM-FC scheme, we consider four main types of attacks against the vehicle platoon scenario. Specifically, (1) the honest-but-curious vehicle cloud server or malicious vehicle platoon clients may impersonate legitimate vehicle platoon users to access the stored data for snooping the data privacy. (2) The malicious vehicle users may collude with the authorized users or cloud server to generate or recover a legitimate secret key for the authorization token generation. (3) The malicious users who launch forgery attacks try to eavesdrop, forge or edit the encrypted data to undermine the data originality. (4) The authorized platoon users may delegate its entire private key to other users for some benefit interests or other purpose. For the consideration of the above attacks, our PDSM-FC should at least reach the following security goals:

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on July 10,2023 at 04:23:22 UTC from IEEE Xplore. Restrictions apply.

- **Platoon data confidentiality**. The encrypted platoon data can only be accessible by authorized platoon users who hold the correct decryption key, while any illegal user or cloud server cannot read or learn the actual contents of encrypted platoon data.
- **Platoon data authenticity**. The encrypted platoon data cannot be forged or edited by malicious system users unless it has legitimate encryption keys.
- **Controllable proxy transformation**. Only the data that have been authorized a legitimate authorization token can be transformed from one-to-one IBE ciphertext format to one-to-many IBBE ciphertext format by the cloud server. Besides, any non-authorization users and the cloud server cannot collaborate with the counterpart to produce a legitimate authorization token.

E. Security Definitions of PDSM-FC

In this section, the formal security definitions are used for realizing the indistinguishability against chosen-plaintext attacks launched by unauthorized vehicle users and the curious vehicle cloud server, the decryption key leakage-resistance against collusion attacks sent by authorized vehicle users and cloud server, and the unforgeability against forgery attacks launched by malicious vehicle users. We define the following security games:

Definition 5 (Indistinguishability Against Chosen-Plaintext Attacks (IND-CPA)): The PDSM-FC scheme is secure against chosen plaintext attacks if $\mathcal{A}$ owns the negligible advantage in wining the following game. The advantage of $\mathcal{A}$ is defined as $Adv_{\mathcal{A}} = |Pr[\beta = \beta'] - 1/2|$.

- **Setup**: An intended identity id^* is picked and sent to the challenger $\mathcal{B}$, which then performs the Setup algorithm to create the public parameters pp and system master secret key msk. Afterwards, pp is given to the adversary $\mathcal{A}$ and msk is kept secretly in its hand.
- **Phases 1 & 2**: The queries as follows can be made by $\mathcal{A}$ to $\mathcal{B}$:
  - Secret key queries $\mathcal{O}_{\mathsf{SKGen}}$: Secret keys for any user with its identity id_i can be queried by $\mathcal{A}$. As a response, $\mathcal{B}$ performs the SKGen algorithm to produce a secret key $\mathsf{sk}_{\mathsf{id}_i}$ for $\mathcal{A}$.
  - Authorization token queries $\mathcal{O}_{\mathsf{Authorize}}$: Authorization tokens can be asked if an identity id_i and a set $\mathcal{S}_i$ of identities are submitted to $\mathcal{B}$. Then, $\mathcal{B}$ checks whether a secret key is produced for id_i. If not, $\mathcal{B}$ first creates a secret key $\mathsf{sk}_{\mathsf{id}_i}$ and then implements $\mathsf{Authorize}(\mathsf{pp}, \mathsf{sk}_{\mathsf{id}}, \mathcal{S})$ to generate an authorization token $\mathsf{ak}_{\mathsf{id}_i \to \mathcal{S}_i}$.
- **Challenge**: $\mathcal{A}$ picks two equal-length messages M_0 and M_1, and sends them to $\mathcal{B}$. It is deserved that the secret key for id^* is not queried and either a search token $\mathsf{ak}_{\mathsf{id}^* \to \mathcal{S}_i}$ or a secret key $\mathsf{sk}_{\mathsf{id}_i}$ is queried. $\mathcal{B}$ flips a coin β to encode M_β, where $\beta \in \{0, 1\}$ and returns a ciphertext ct to $\mathcal{A}$.
- **Guess**: A guess $\beta' \in \{0, 1\}$ is given to $\mathcal{B}$ and $\mathcal{A}$ wins this game if $\beta = \beta'$.

Definition 6 (Decryption Key Leakage-Resistance Against Collusion Attacks (LR-CA)): The PDSM-FC scheme is secure against collusion attacks if $\mathcal{A}$ has the negligible advantage in wining the following game. The advantage of $\mathcal{A}$ is defined as $Adv_{\mathcal{A}} = |Pr[sk'_{id} = sk_{id^*}] - 1/2|$.

- **Setup**: $\mathcal{B}$ carries out the Setup algorithm to create the public parameters pp and system master secret key msk. Then, pp is given to $\mathcal{A}$ and msk is kept secretly in its hand.
- **Phases 1 & 2**: The identity id^* is sent and the queries as below can be made by $\mathcal{A}$ to $\mathcal{B}$:
  - Secret key queries $\mathcal{O}_{\mathsf{SKGen}}$: The same as that of **Definition 5**.
  - Authorization token queries $\mathcal{O}_{\mathsf{Authorize}}$: The same as that of **Definition 5**.
- **Challenge**: $\mathcal{A}$ outputs a secret key $\mathsf{sk}'_{\mathsf{id}}$ related to id^*.
- **Guess**: $\mathcal{A}$ wins this game if the following restrictions are satisfied: $\mathsf{sk}'_{\mathsf{id}} = \mathsf{sk}_{\mathsf{id}^*}$ and the secret key id^* is never queried.

Definition 7 (Unforgeability Against Forgery Attacks (UNF-FA)): The PDSM-FC scheme achieves authenticity if the BDH instance holds.

- **Setup**: $\mathcal{B}$ conducts the Setup algorithm to create the public parameters pp and system master secret key msk. Then, it sends pp to $\mathcal{A}$ and keeps msk in its hand.
- **Oracle Queries**: The queries of encryption keys and secret keys are issued by $\mathcal{A}$, and B performs the EKGen & SKGen algorithms to create and send $\mathsf{sk}_{\mathsf{id}'}$ and $\mathsf{ek}_{\mathsf{id}'}$. To generate the corresponding keys, the following hash functions H_0, H_1, H_4 are formalized as random oracles. Specifically:
  - Encryption key queries $\mathcal{O}_{\mathsf{EKGen}}$: Let id_i denote the input of the oracle EKGen. $\mathcal{B}$ performs the EKGen algorithm and returns $\mathsf{ek}_{\mathsf{id}'}$.
  - Secret key queries $\mathcal{O}_{\mathsf{SKGen}}$: Almost the same as that of **Definition 5**. Note that only partial secret keys are required to produce in this game.
- **Forgery**: $\mathcal{A}$ forges a valid forgery $(\mathsf{ct}, \mathsf{id}, \mathsf{id}^*)$ if for $\mathsf{ek}_{\mathsf{id}'}$ obtained by $\mathcal{A}$ it satisfies that $\mathsf{id}' = \mathsf{id}^*$, and moreover id is not held by $\mathcal{A}$.

IV. CONSTRUCTION

This section first presents the details of our concrete construction. Then, the deployment of PDSM-FC in the vehicle platoon merging application is elaborately described to achieve secure cross-platoon data sharing.

A. Concrete Construction

Our construction is invented based on the idea of matchmaking encryption approach and cross-domain ciphertext transformation technique. The following is the details of the designed algorithms:

- Setup $(1^\lambda, \ell)$: Input a security parameter 1^λ and the allowed maximum number ℓ of users to simultaneously access some data, it first chooses a bilinear map group $\mathbb{B} = (\mathbb{G}_0, \mathbb{G}_1, e, p)$, where $e : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_1$ and $\mathbb{G}_0, \mathbb{G}_1$ are two cyclic groups with the same prime order p. Next, it picks a random generator $g \in \mathbb{G}_0$, $\theta, x, y \in \mathbb{Z}_p^*$, $v, u, \mathsf{P}, \mathsf{Q} \in \mathbb{G}_0$

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on July 10,2023 at 04:23:22 UTC from IEEE Xplore. Restrictions apply.

and computes $\mathsf{P}_0 = \mathsf{P}^x, \mathsf{Q}_0 = \mathsf{Q}^y, g_1 = g^\theta, v^\theta, u^\theta, u^{\theta^2}, \ldots, u^{\theta^\ell}$. After that, it selects five hash functions $\mathsf{H}_0 : \{0,1\}^* \to \mathbb{G}_0, \mathsf{H}_1 : \{0,1\}^* \to \mathbb{G}_0$, $\mathsf{H}_2 : \{0,1\}^* \to \mathbb{Z}_p^*$, $\mathsf{H}_3 : \mathbb{G}_1 \to \mathbb{G}_0$ and $\mathsf{H}_4 : \mathbb{G}_1 \to \mathbb{G}_1$. It finally outputs a system public parameter $\mathsf{pp} = (g_1, v, v^\theta, u, u^\theta, u^{\theta^2}, \ldots, u^{\theta^\ell}, \mathsf{P}, \mathsf{P}_0, \mathsf{Q}, \mathsf{Q}_0, \{\mathsf{H}_i\}_{i\in[0,4]}, e(g,u))$ and a system master secret key $\mathsf{msk} = (g, \theta, x, y)$.

- EKGen (id^*, msk): Input pp, msk and an identity id^*, it outputs an encryption key $\mathsf{ek}_{\mathsf{id}^*} = \mathsf{H}_1(\mathsf{id}^*)^y$.
- SKGen (pp, msk, id): Input pp, msk and an identity id, it outputs a secret key $\mathsf{sk}_{\mathsf{id}} = (\mathsf{sk}^0_{\mathsf{id}}, \mathsf{sk}^1_{\mathsf{id}}, \mathsf{sk}^2_{\mathsf{id}}, \mathsf{sk}^3_{\mathsf{id}})$, where $\mathsf{sk}^0_{\mathsf{id}} = g^{\frac{1}{\theta+\mathsf{H}_2(\mathsf{id})}}$, $\mathsf{sk}^1_{\mathsf{id}} = \mathsf{H}_0(\mathsf{id})^x, \mathsf{sk}^2_{\mathsf{id}} = \mathsf{H}_0(\mathsf{id})^y, \mathsf{sk}^3_{\mathsf{id}} = \mathsf{H}_0(\mathsf{id})$.
- Encrypt (pp, M, id, $\mathsf{ek}_{\mathsf{id}^*}$): Input pp, $\mathsf{ek}_{\mathsf{id}^*}$, the message M and an identity id, it first chooses $r_0, r_1 \in \mathbb{Z}_p^*$ and computes $\mathsf{R}_0 = \mathsf{P}^{r_0}, \mathsf{R}_1 = \mathsf{P}^{r_1}$. Next, it computes $k_R = e(\mathsf{H}_0(\mathsf{id}), \mathsf{P}_0^{r_1})$ and $k_S = e(\mathsf{H}_0(\mathsf{id}), \mathsf{R}_0 \cdot \mathsf{ek}_{\mathsf{id}^*})$. It also computes $c_0 = M \cdot e(g,u)^{r_1} \oplus \mathsf{H}_4(k_R) \oplus \mathsf{H}_4(k_S)$, $c_1 = u^{r_1(\theta+\mathsf{H}_2(\mathsf{id}))}$, $c_2 = v^{r_1(\theta+\mathsf{H}_2(\mathsf{id}))}$. Finally, it outputs an IBE ciphertext $\mathsf{ct}_{\mathsf{id}} = (c_0, c_1, c_2, \mathsf{R}_0, \mathsf{R}_1)$.
- Authorize (pp, $\mathsf{sk}_{\mathsf{id}}$, $\mathcal{S}$): Input pp, $\mathsf{sk}_{\mathsf{id}}$ and the identity set $\mathcal{S}$ of data receivers, where $\mathcal{S} = \{\mathsf{id}_i\}_{i=1}^{n\le\ell}$, it first picks $s, t \in \mathbb{Z}_p$ and computes $s_1 = g_1^{-s}, s_2 = u^{s\prod_{i=1}^{n}(\theta+\mathsf{H}_2(\mathsf{id}_i))}$, $s_3 = \mathsf{H}_3(e(g,u)^s) \cdot u^t$, $s_4 = \mathsf{sk}^0_{\mathsf{id}} \cdot v^{-t}$, $s_5 = \mathsf{sk}^1_{\mathsf{id}}$, $s_6 = \mathsf{sk}^2_{\mathsf{id}}$, $s_7 = \mathsf{sk}^3_{\mathsf{id}}$. Finally, it outputs an authorized token $\mathsf{at}_{\mathsf{id}\to\mathcal{S}} = (s_1 \ldots, s_7)$.
- Transform (pp, $\mathsf{at}_{\mathsf{id}\to\mathcal{S}}$, $\mathsf{ct}_{\mathsf{id}}$): Input pp, $at_{\mathsf{id}\to\mathcal{S}} = (s_1 \ldots, s_7)$ and $\mathsf{ct}_{\mathsf{id}} = (c_0, c_1, c_2, \mathsf{R}_0, \mathsf{R}_1)$, it first computes $k_R = e(s_5, R_1), k_S = e(s_6, \mathsf{H}_1(\mathsf{id}^*)) \cdot e(s_7, R_0)$ and $c'_0 = c_0 \oplus \mathsf{H}_4(k_R) \oplus \mathsf{H}_4(k_S)$. Next, it sets $c'_1 = s_1, c'_2 = s_2, c'_3 = s_3, c'_4 = c_2$ and computes $c'_5 = c'_0/e(c_1, s_4) = M \cdot e(u^{r_1(\theta+\mathsf{H}_2(\mathsf{id}))}, v^t)$. Finally, it outputs a transformed IBBE ciphertext $\mathsf{ct}_{\mathcal{S}} = (c'_1, \ldots, c'_5)$.
- Decrypt (pp, $\mathsf{ct}_{\mathsf{id}}/\mathsf{ct}_{\mathcal{S}}$, $\mathsf{sk}_{\mathsf{id}'}$): Input pp, $\mathsf{ct}_{\mathsf{id}}/\mathsf{ct}_{\mathcal{S}}$ and $\mathsf{sk}_{\mathsf{id}'}$, it outputs M if $\mathsf{id} = \mathsf{id}'$ or $\mathsf{id}' \in \mathcal{S}$, and otherwise it returns $\perp$. Specifically,
  - If the ciphertext is an IBE ciphertext $\mathsf{ct}_{\mathsf{id}} = (c_0, c_1, c_2, \mathsf{R}_0, \mathsf{R}_1)$ and $\mathsf{id} = \mathsf{id}'$, it first computes $k_R = e(\mathsf{sk}^1_{\mathsf{id}}, R_1), k_S = e(\mathsf{sk}^2_{\mathsf{id}}, \mathsf{H}_1(\mathsf{id}^*)) \cdot e(\mathsf{sk}^3_{\mathsf{id}}, R_0)$ and $c'_0 = c_0 \oplus \mathsf{H}_4(k_R) \oplus \mathsf{H}_4(k_S)$. Next, it recovers M by computing $M = c'_0/e(\mathsf{sk}^1_{\mathsf{id}}, c_1)$.
  - If the ciphertext is an IBBE ciphertext $\mathsf{ct}_{\mathcal{S}} = (c'_0, \ldots, c'_5)$ and $\mathsf{id}_i \in \mathcal{S}$, it first computes $\mathsf{A} = (e(c'_1, u^{\Delta_{i,\mathcal{S}}(\theta)}) \cdot e(\mathsf{sk}^0_{\mathsf{id}_i}, c'_2))^{\frac{1}{\prod_{k=1,k\neq i}^{n}\mathsf{H}_2(\mathsf{id}_k)}}$ and $\mathsf{B} = u^t = c'_3/\mathsf{H}_3(\mathsf{A})$, where $\Delta_{i,\mathcal{S}}(\theta) = \frac{1}{\theta}(\prod_{k=1,k\neq i}^{n}(\theta + \mathsf{H}_2(\mathsf{id}_k)) - \prod_{k=1,k\neq i}^{n}\mathsf{H}_2(\mathsf{id}_k))$. Next, it recovers M by counting $M = c'_5/e(\mathsf{B}, c'_4)$.

B. Deployment of PDSM-FC in Vehicle Platoon Merging

Consider the scenario that two vehicle platoons (A and B) plan to merge into one platoon. The PL of Platoon A (PL^A) is still the PL of the new platoon. Four phases are involved in the the process of secure data sharing for completing the platoon merging in PDSM-FC: System Initialization, Platoon Data Uploading, Platoon Data Transformation and Platoon Data Recovery.

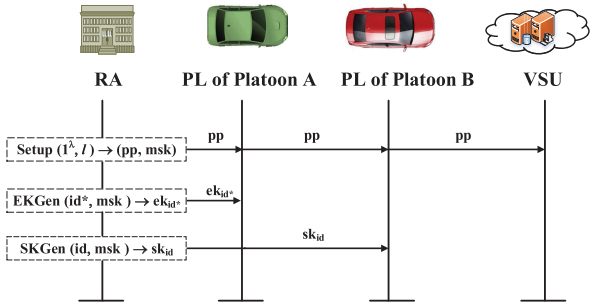


Fig. 2. System initialization.

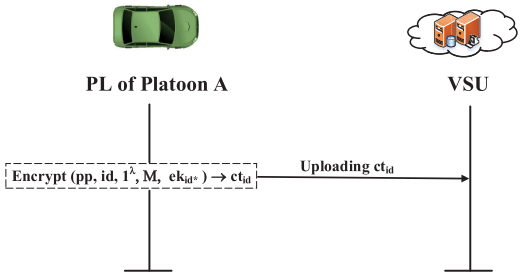


Fig. 3. Platoon data uploading.

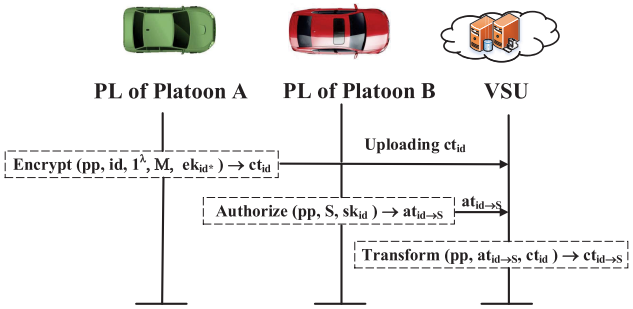


Fig. 4. Platoon data transformation.

1) System Initialization: Fig. 2 shows the initialization phase. Specifically, RA conducts the **Setup** algorithm to generate and assign the public parameter to each system entity. Besides, RA runs the **EKGen** and **SKGen** algorithms to create an encryption key and a secret key for each vehicle including the platoon leaders and followers.

2) Platoon Data Uploading: Fig. 3 depicts the data uploading phase. PL^A runs the **Encrypt** algorithm to encrypt its platoon data, such as platoon location, location size, platoon identity, etc., and then forwards them to the vehicle cloud server unit (VSU) for sharing with PL^B.

3) Platoon Data Transformation: Fig. 4 indicates the data transformation phase. PL^B runs the **Authorize** algorithm to blind its secret key, and then forwards it to the VSU, which then runs the **Transform** algorithm to convert the previous IBE ciphertext into a new IBBE ciphertext, such that PFs of

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on July 10,2023 at 04:23:22 UTC from IEEE Xplore. Restrictions apply.

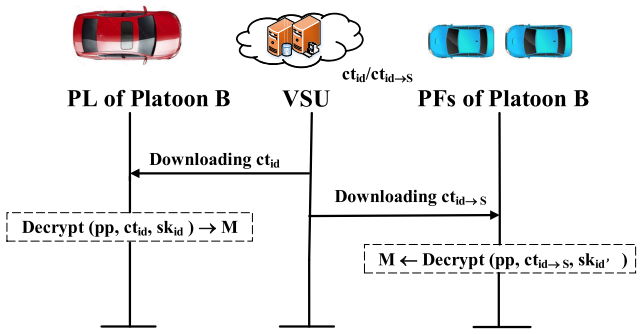


Fig. 5. Platoon data recovery.

Platoon B can use their own secret key to access the same platoon data.

4) Platoon Data Recovery: Fig. 5 presents the data recovery phase. For PL^B, it needs to run the first step of the **Decrypt** algorithm to decrypt the IBE ciphertext. For the PFs of Platoon B, they require to perform the second step of the **Decrypt** algorithm to decrypt the IBBE ciphertext. After getting the specific data of Platoon A, Platoon B can implement the merging task with Platoon A to form a new platoon.

V. Security Proofs and Analysis

In this section, we first prove the correctness of PDSM-FC and then give the strict security proofs to indicate the strong security of PDSM-FC. Finally, we elaborate on the security analysis to demonstrate that PDSM-FC is invulnerable to impersonation attacks, collusion attacks, and forgery attacks.

A. Security Proofs

Theorem 1: For both IBE and IBBE ciphertexts, a user having correct secret keys succeeds in deciphering the ciphertext.

Proof: This theorem can be correctly proved via the following cases. Specifically,

- for an IBE ciphertext $\mathsf{ct_{id}} = (c_0, c_1, c_2, \mathsf{R}_0, \mathsf{R}_1)$, a user with $\mathsf{sk_{id}} = (\mathsf{sk}^0_{\mathsf{id}}, \mathsf{sk}^1_{\mathsf{id}}, \mathsf{sk}^2_{\mathsf{id}}, \mathsf{sk}^3_{\mathsf{id}})$ performs the following equations to recover M:

$$\begin{aligned} k'_R &= e(\mathsf{sk}^1_{\mathsf{id}}, R_1) = e(\mathsf{H}_0(\mathsf{id})^x, \mathsf{P}^{r_1}) \\ &= e(\mathsf{H}_0(\mathsf{id}), \mathsf{P}_0^{r_1}) = k_R, \\ k'_S &= e(\mathsf{sk}^2_{\mathsf{id}}, \mathsf{H}_1(\mathsf{id}^*)) \cdot e(\mathsf{sk}^3_{\mathsf{id}}, R_0) \\ &= e(\mathsf{H}_0(\mathsf{id})^y, \mathsf{H}_1(\mathsf{id}^*)) \cdot e(\mathsf{H}_0(\mathsf{id}), \mathsf{P}^{r_0}) \\ &= e(\mathsf{H}_0(\mathsf{id}), (\mathsf{H}_1(\mathsf{id}^*))^y) \cdot e(\mathsf{H}_0(\mathsf{id}), \mathsf{P}^{r_0}) \\ &= e(\mathsf{H}_0(\mathsf{id}), \mathsf{R}_0 \cdot \mathsf{ek}_{\mathsf{id}^*}) = k_S, \\ c'_0 &= c_0 \oplus k'_R \oplus k'_S = M \cdot e(g,u)^{r_1}, \\ M &= c'_0 / e((\mathsf{sk}^0_{\mathsf{id}}, c_1). \end{aligned}$$

- for an IBBE ciphertext $\mathsf{ct}_{\mathcal{S}} = (c'_1, \ldots, c'_5)$, a user with $\mathsf{sk}^0_{\mathsf{id}_i}$ performs the following equations to recover M if $\mathsf{id} \in \mathcal{S}$:

$$\begin{aligned} \mathsf{A} &= (e(c'_1, u^{\Delta_{i,\mathcal{S}}(\theta)}) \cdot e(\mathsf{sk}^0_{\mathsf{id}_i}, c'_2))^{\frac{1}{\prod_{k=1,k\neq i}^{n} \mathsf{H}_2(\mathsf{id}_k)}} \\ &= (e(g_1^{-s}, u^{\frac{1}{\theta}(\prod_{k=1,k\neq i}^{n}(\theta+\mathsf{H}_2(\mathsf{id}_i)) - \prod_{k=1,k\neq i}^{n}\mathsf{H}_2(\mathsf{id}_i))}) \\ &\quad \cdot e(g^{\frac{1}{\theta+\mathsf{H}_2(\mathsf{id}_i)}}, g^{s\prod_{k=1,k\neq i}^{n}(\theta+\mathsf{H}_2(\mathsf{id}_i))})^{\frac{1}{\prod_{k=1,k\neq i}^{n}\mathsf{H}_2(\mathsf{id}_k)}}) \\ &= e(g^s, u), \\ \mathsf{B} &= u^t = c'_3/\mathsf{H}_3(\mathsf{A}), \quad M = c'_5/e(\mathsf{B}, c'_4). \end{aligned}$$

■

From the above process, it's easy to conclude that this theorem is correct.

Theorem 2: PDSM-FC can achieve leakage-resistance of decryption key against collusion attacks as well as realize traceability if the modified ℓ-SDH assumption holds.

Proof: Assume that a secret key of a delegator with id^* is retrieved by an adversary $\mathcal{A}$. Then an algorithm $\mathcal{B}$ can be constructed to simulate as a challenger to solve the modified ℓ-SDH assumption. Given an instance $(g_0, g_0^{\theta}, g_0^{\theta^2}, \ldots, g_0^{\theta^\ell}, \rho)$ of the modified ℓ-SDH assumption, the goal of $\mathcal{B}$ is to get the result $g_0^{1/(\theta+\rho)}$.

- **Setup**: A polynomial $f(x) = \prod_{i=1}^{\ell-1}(x + t_i)$ with randomly selected values $t_i \in \mathbb{Z}_p^*$ is first defined by $\mathcal{B}$. It is easy to expand the above polynomial to obtain $f(x) = \sum_{i=0}^{\ell-1} a_i x^i$, where a_i are the polynomial f's coefficients. After that, $\mathcal{B}$ chooses a random $\tau \in \mathbb{Z}_p$, computes $g = \prod_{i=0}^{q-1}(g_0^{\theta^i})^{a_i\tau}$ and $g^{\theta} = \prod_{i=0}^{q-1}(g_0^{\theta^{i+1}})^{a_i\tau}$. Correspondingly, $g = g_0^{\tau f(\theta)}$ and $g_1 = g_0^{\tau\theta f(\theta)}$ are implicitly set. $\mathcal{B}$ also randomly selects $\varphi, \phi \in \mathbb{Z}_p$, computes $v = g_0^{\varphi}$, $v^{\theta} = g_0^{\theta\varphi}$ and $u = g_0^{(\theta+\rho)\phi}$. Hence, $e(g,u) = e(g_0^{\tau f(\theta)}, g_0^{(\theta+\rho)\phi})$ and $u^{\theta^i} = g_0^{\phi\theta^{i+1}} \cdot g_0^{\rho\phi\theta^i} = (g_0^{(\theta+\rho)\phi})^{\theta^i}$, where $i \in [1, \ell]$. Finally, the public parameter $\mathsf{pp} = (g_1, v, v^{\theta}, u, u^{\theta}, u^{\theta^2}, \ldots, u^{\theta^\ell}, e(g,u))$ is returned to $\mathcal{A}$ by $\mathcal{B}$.
- **Phases 1 & 2**: The following hash functions are modeled by $\mathcal{B}$ as oracle models: $\mathsf{H}_0 : \{0,1\}^* \to \mathbb{G}_0$, $\mathsf{H}_2 : \{0,1\}^* \to \mathbb{Z}_p^*$, and $\mathsf{H}_3 : \mathbb{G}_1 \to \mathbb{G}_0$. $\mathcal{B}$ starts by building a table T_0 of tuples $(\mathsf{id}_i, \rho_i, \mathsf{sk}_{\mathsf{id}_i})$. For this hash on id_i, if table T_0 contains id_i, then it returns ρ_i. Otherwise, it returns $\mathsf{H}_2(\mathsf{id}_i) = \rho_i$. $\mathcal{A}$ gives an identity id^* and adaptively makes the following queries to $\mathcal{B}$:
  - $\mathcal{O}_{\mathsf{SKGen}}$: $\mathcal{A}$ gives an identity $\mathsf{id}_i \neq \mathsf{id}^*$ for making requests of secret key generation for id_i. If the queried secret key for id_i is done by $\mathcal{A}$, the corresponding secret key $\mathsf{sk}_{\mathsf{id}_i}$ in the table list T_0 is returned to $\mathcal{A}$. Otherwise, $\mathcal{B}$ uses the corresponding ρ_i to produce the secret key as below. First, it defines $f'(x) = f(x)/(x + \rho_i) = \prod_{j=1,j\neq i}^{\ell-1}(x + \rho_j)$. Also, it is easy to expand $f'(x)$ to obtain $f'(x) = \sum_{j=0}^{q-2} b_j x^j$ with its coefficients b_j. After that, $\mathcal{B}$ computes $\mathsf{sk}^0_{\mathsf{id}_i} = \prod_{j=0}^{\ell-2}(g_0^{\theta^j})^{b_j\tau}$, and then gets $\mathsf{sk}^0_{\mathsf{id}_i} = g_0^{\tau f'(\theta)} = g^{1/(\theta+\mathsf{H}_2(\mathsf{id}_i))}$. Finally, $\mathcal{B}$ returns $\mathsf{sk}_{\mathsf{id}_i}$ and updates the table T_0 with $\mathsf{sk}^0_{\mathsf{id}_i}$ for identity id_i. If $\mathcal{A}$ have queried neither the secret key $\mathsf{sk}^0_{\mathsf{id}_i}$, nor the hash value of id_i, then $\mathsf{H}_2(\mathsf{id}_i) = \rho_i$ is set. Thus, the secret key $\mathsf{sk}^0_{\mathsf{id}_i}$ is computed and the table T_0 is also updated as before. Besides, $\mathcal{B}$ randomly picks $x', y' \in \mathbb{Z}_p^*$ and sets $\mathsf{sk}^1_{\mathsf{id}} = \mathsf{H}_0(\mathsf{id})^{x'}$, $\mathsf{sk}^2_{\mathsf{id}} = \mathsf{H}_0(\mathsf{id})^{y'}$, $\mathsf{sk}^3_{\mathsf{id}} = \mathsf{H}_0(\mathsf{id})$.

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on July 10,2023 at 04:23:22 UTC from IEEE Xplore. Restrictions apply.

– $\mathcal{O}_{\mathsf{Authorize}}$: $\mathcal{A}$ makes requests to generate an authorized token $at_{\mathsf{id}^*\to\mathcal{S}_i}$ from id^* to the identity set $\{\mathcal{S}_i\}_{j=1}^n$. To produce the authorized token, $\mathcal{B}$ sets $\mathsf{H}_2(\mathsf{id}^*)=\rho$ and records $(\mathsf{id}^*,\rho,*)$ in T_0. Next, $\mathcal{B}$ chooses random values $s,t'\in\mathbb{Z}_p^*$ and counts $at_{\mathsf{id}^*\to\mathcal{S}_i}=(s_1\ldots,s_7)$, where $s_1=g_1^{-s}$, $s_2=u^s\prod_{i=1}^n(\theta+\mathsf{H}_2(\mathsf{id}_j))$, $s_3=\mathsf{H}_3(e(g,u)^s)\cdot((g_0)^{\frac{\tau\phi}{\varphi}f(\theta)}\cdot u^{t'})$, $s_4=\upsilon^{-t'}$, $s_5=\mathsf{H}_0(\mathsf{id}^*)^x$, $s_6=\mathsf{H}_0(\mathsf{id}^*)^y$, $s_7=\mathsf{H}_0(\mathsf{id}^*)$. Note that set $t=t'+\frac{\tau}{\varphi}\cdot\frac{f(\theta)}{\theta+\rho}$, it is simple to get $u^t=(g_0^{(\theta+\rho)\phi})^{t'+\frac{\tau}{\varphi}\cdot\frac{f(\theta)}{\theta+\rho}}=u^{t'}\cdot g_0^{\frac{\tau\phi}{\varphi}f(\theta)}$ and $\mathsf{sk}^1_{\mathsf{id}^*}\cdot\upsilon^{-t'}=\mathsf{sk}^1_{\mathsf{id}^*}\cdot g_0^{\varphi(-t'-\frac{\tau}{\varphi}\cdot\frac{f(\theta)}{\theta+\rho})}=\mathsf{sk}^1_{\mathsf{id}^*}\cdot\mathsf{sk}_{\mathsf{id}^*}^{-1}\cdot g_0^{-\varphi t'}=\upsilon^{-t'}$. From the above descriptions, it is easy to get $s_3=\mathsf{H}_3(e(g,u)^s)\cdot u^t$, $s_4=\mathsf{sk}^1_{\mathsf{id}^*}\cdot\upsilon^{-t}$. Hence, $at_{\mathsf{id}^*\to\mathcal{S}_i}=(s_1\ldots,s_7)$ can be set as an authorized token.

- **Challenge**: $\mathcal{A}$ returns a part of the valid secret keys $\mathsf{sk}^1_{\mathsf{id}'}$ for identity id^*. Since $e(u^\theta u^\rho,\mathsf{sk}^1_{\mathsf{id}'})=e(g,u)$, where $\rho=\mathsf{H}_2(\mathsf{id}^*)$, it is therefore simple to get $\mathsf{sk}^1_{\mathsf{id}'}=g^{\frac{1}{\theta+\rho}}=g_0^{\frac{\theta f(\theta)}{\theta+\rho}}$. For the ratio $f(\theta)/(\theta+\rho)$ in $\mathsf{sk}^1_{\mathsf{id}'}$, we reset the polynomial f as $f(x)=(x+\rho)\lambda(x)+\varepsilon$, where $\lambda(x)=\sum_{i=0}^{\ell-2}\lambda_i x^i$ and ε is a constant value from $\mathbb{Z}_p^*$. Further, the ratio $f(\theta)/(\theta+\rho)$ can be set as $f(\theta)/(\theta+\rho)=\sum_{i=0}^{\ell-2}\lambda_i x^i+\varepsilon/(x+\rho)$. Therefore, $\mathsf{sk}^1_{\mathsf{id}'}$ can be re-denoted as $\mathsf{sk}^1_{\mathsf{id}'}=g_0^{\tau(\sum_{i=0}^{\ell-2}\lambda_i\theta^i+\varepsilon/(\theta+\rho))}$, where $\varepsilon\neq 0$ due to the fact that $f(x)$ cannot be divided by $(x+\rho)$.
- **Guess**: $\mathcal{B}$ outputs $\mathcal{R}$ as a solution to the instance of the modified ℓ-SDH assumption, where $\mathcal{R}=((\mathsf{sk}^1_{\mathsf{id}'})^{\tau^{-1}}\prod_{i=0}^{\ell-2}(g_0^{\theta^i})^{-\lambda_i})^{\varepsilon^{-1}}=(g_0^{\sum_{i=0}^{\ell-2}\lambda_i\theta^i}\cdot g_0^{-\sum_{i=0}^{\ell-2}\lambda_i\theta^i}\cdot g_0^{\frac{\varepsilon}{\theta+\rho}})^{\varepsilon^{-1}}=g_0^{\frac{1}{\theta+\rho}}$. Hence, if there exists an adversary that has some advantages in breaking the security of the proposed scheme, then the modified ℓ-SDH assumption can be solved with the same advantage. ■

Theorem 3: PDSM-FC is secure against chosen plaintext attacks if the GDDHE assumption holds.

Proof: Suppose that adversary $\mathcal{A}$ can break our PDSM-FC, then a challenger $\mathcal{B}$ can be constructed via interacting with $\mathcal{A}$ to solve the GDDHE assumption. Given an instance $(g_0^\theta,\ldots,g_0^{\theta^{\ell-1}},g_0^{\theta\mathcal{P}(\theta)},g_0^{r_1\theta\mathcal{P}(\theta)},u_0,u_0^\theta,\ldots,u_0^{\theta^{2k}},u_0^{r_1\theta\mathcal{Q}(\theta)},g_0,\mathcal{Z})$ of GDDHE assumption, the goal of $\mathcal{A}$ is to decide $\mathcal{Z}=e(g_0,u_0)^{r_1\mathcal{P}(\theta)}$, where two different generators of cyclic group $\mathbb{G}_0$ are set as g_0,u_0 and two polynomials $\mathcal{P}(x)=\prod_{i=1}^{\ell}(x+\rho_i)$, $\mathcal{Q}(x)=\prod_{i=\ell+1}^{\ell+k}(x+\rho_i)$ own various roots.

- **Setup**: $\mathcal{A}$ picks a challenge id^* and submits it to $\mathcal{B}$. The maximum number of identities who are authorized to access the data are set as k. In this way, $\mathcal{B}$ sets $g_1=\mathsf{P}=g_0^{\theta\mathcal{P}(\theta)}$, $u=u_0^{\prod_{i=\ell+2}^{\ell+k}(x+\rho_i)}$, $u^\theta=u_0^{\theta\prod_{i=\ell+2}^{\ell+k}(x+\rho_i)}$, …, $u^{\theta^k}=u_0^{\theta^k\prod_{i=\ell+2}^{\ell+k}(x+\rho_i)}$, $e(g,u)=e(g_0,u_0)^{\mathcal{P}(\theta)\prod_{i=\ell+2}^{\ell+k}(x+\rho_i)}$. Besides, $\mathcal{B}$ randomly selects $\eta\in\mathbb{Z}_p^*$ and computes $\upsilon=u^\eta$, $\upsilon^\theta=u^{\theta\eta}$. After that, the following hash functions are also modeled as oracle models: $\mathsf{H}_0:\{0,1\}^*\to\mathbb{G}_0$, $\mathsf{H}_1:\{0,1\}^*\to\mathbb{G}_0$, $\mathsf{H}_2:\{0,1\}^*\to\mathbb{Z}_p^*$, $\mathsf{H}_3:\mathbb{G}_1\to\mathbb{G}_0$ and $\mathsf{H}_4:\mathbb{G}_1\to\mathbb{G}_1$. Specifically, a table T_0 of tuples $(\mathsf{id}_i,\rho_i,\mathsf{sk}_{\mathsf{id}_i})$ is created. For this hash on id_i, if table T_0 contains id_i, then it returns ρ_i. Otherwise, it returns $\mathsf{H}_2(\mathsf{id}_i)=\rho_i$. Besides, $\mathcal{B}$ randomly selects $x',y'\in\mathbb{Z}_p^*$ and computes $\mathsf{P}_0=\mathsf{P}^{x'}$. Finally, $\mathcal{B}$ sends the initialized public parameter $\mathsf{pp}=(g_1,\upsilon,\upsilon^\theta,u,u^\theta,\ldots,u^{\theta^k},\mathsf{P},\mathsf{P}_0,e(g,h),\{\mathsf{H}_i\}_{i\in[0,4]})$.

  Note that the master secret key $\mathsf{msk}=(g_0^{\mathcal{P}(\theta)},\theta)$ is unknown to $\mathcal{B}$.
- **Phases 1 & 2**: $\mathcal{A}$ adaptively makes the following queries to $\mathcal{B}$:
  – $\mathcal{O}_{\mathsf{SKGen}}$: $\mathcal{A}$ sends queries of the secret key for an identity $\mathsf{id}_i\neq\mathsf{id}^*$. If the secret key has been queried before, then the corresponding secret key is returned to $\mathcal{A}$. Otherwise, $\mathcal{B}$ checks the table T_0 to find the hash value $\mathsf{H}_2(\mathsf{id}_i)=\rho_i$, computes $\mathsf{sk}^0_{\mathsf{id}_i}=g^{\frac{1}{\theta+\mathsf{H}_2(\mathsf{id}_i)}}=g_0^{\frac{\mathcal{P}(\theta)}{\theta+\rho_i}}$ and sets $\mathsf{sk}^1_{\mathsf{id}}=\mathsf{H}_0(\mathsf{id})^x$, $\mathsf{sk}^2_{\mathsf{id}}=\mathsf{H}_0(\mathsf{id})^y$, $\mathsf{sk}^3_{\mathsf{id}}=\mathsf{H}_0(\mathsf{id})$, where $x,y\in\mathbb{Z}_p^*$ are picked. After that, $\mathcal{B}$ gives the secret key back to $\mathcal{A}$ and updates the table T_0. If neither the secret key $\mathsf{sk}^0_{\mathsf{id}_i}$, nor the hash value of id_i is queried by $\mathcal{A}$, then $\mathsf{H}_2(\mathsf{id}_i)=\rho_i$ is set. Thus, the secret key $\mathsf{sk}^0_{\mathsf{id}_i}$ is computed and the table T_0 is also updated as before.
  – $\mathcal{O}_{\mathsf{Authorize}}$: $\mathcal{A}$ makes requests to generate an authorized token $at_{\mathsf{id}^*\to\mathcal{S}_i}$ from id^* to the identity set $\{\mathcal{S}_i\}_{j=1}^n$. To produce the authorized token $at_{\mathsf{id}^*\to\mathcal{S}_i}$, the secret key is required to be produced. Here a table list $L=(\mathsf{id}_i,\mathcal{S}_i,at_{\mathsf{id}_i\to\mathcal{S}_i})$ is maintained to store the information of the authorized token. Two cases are discussed as below: If $\mathsf{id}_i\neq\mathsf{id}^*$, $\mathcal{B}$ produces the secret key as above. If $\mathsf{id}_i=\mathsf{id}^*$, $\mathsf{sk}_{\mathsf{id}_i}$ cannot be produced due to the absence of $(\theta+\rho_{\ell+1})$ in $\mathcal{P}(\theta)$. Specifically, the authorized token $at_{\mathsf{id}_i\to\mathcal{S}_i}=(s_1\ldots,s_7)$ can be set as follows.
    * $\mathsf{id}_i\neq\mathsf{id}^*$: $\mathcal{B}$ chooses random values $s,t\in\mathbb{Z}_p^*$ and counts $at_{\mathsf{id}^*\to\mathcal{S}_i}=(s_1\ldots,s_7)$, where $s_1=g_1^{-s}$, $s_2=u^s\prod_{i=1}^n(\theta+\mathsf{H}_2(\mathsf{id}_j))$, $s_3=\mathsf{H}_3(e(g,u)^s)\cdot u^t$, $s_4=\mathsf{sk}^1_{\mathsf{id}_i}\cdot\upsilon^{-t}$, $s_5=\mathsf{H}_0(\mathsf{id}_i)^{x'}$, $s_6=\mathsf{H}_0(\mathsf{id}_i)^{y'}$, $s_7=\mathsf{H}_0(\mathsf{id}_i)$.
    * $\mathsf{id}_i=\mathsf{id}^*$: $\mathcal{B}$ outputs a random authorization token by randomly choosing $s,t\in\mathbb{Z}_p$, $s_4\in\mathbb{G}_0$ and setting $s_1=g_1^{-s}$, $s_2=u^s\prod_{i=1}^n(\theta+\mathsf{H}_2(\mathsf{id}_j))$, $s_3=\mathsf{H}_3(e(g,u)^s)\cdot u^t$, $s_5=\mathsf{H}_0(\mathsf{id}_i)^{x'}$, $s_6=\mathsf{H}_0(\mathsf{id}_i)^{y'}$, $s_7=\mathsf{H}_0(\mathsf{id}_i)$.

  For the above cases, $\mathcal{B}$ produces the authorized token $at_{\mathsf{id}_i\to\mathcal{S}_i}$ and updates the latest list L with recording $(\mathsf{id}_i,\mathcal{S}_i,at_{\mathsf{id}_i\to\mathcal{S}_i})$.
- **Challenge**: $\mathcal{A}$ gives two equal-length messages $\{M_i\}_{i\in\{0,1\}}$ to $\mathcal{B}$. If a record $(\mathsf{id}^*,\mathcal{S}_i,at_{\mathsf{id}^*\to\mathcal{S}_i})$ exists in L and a record $(\mathsf{id}_i,\mathsf{sk}_{\mathsf{id}_i})$ exists for all $\mathsf{id}_i\in\mathcal{S}_i$, $\mathcal{B}$ aborts the game. Otherwise, $\mathcal{B}$ picks a random $\beta\in\{0,1\}$, $r_0\in\mathbb{Z}_p^*$, computes $\mathsf{R}_0=\mathsf{P}^{r_0}$ and sets, $\mathsf{R}_1=\mathsf{P}^{r_1}=g_0^{r_1\theta\mathcal{P}(\theta)}$. Then, it continues to perform $k_R=e(\mathsf{H}_0(\mathsf{id}^*),\mathsf{R}_1^{x'})$ and $k_S=e(\mathsf{H}_0(\mathsf{id}^*),\mathsf{R}_0\cdot\mathsf{H}_1(\mathsf{id}')^{y'})$, where id' is publicly chosen. Besides, it also calculates $c_0=M_b\cdot\mathcal{Z}^{\prod_{i=\ell+2}^{\ell+k}\rho_i}\cdot e(g_0^{r_1\theta\mathcal{P}(\theta)},u_0^{\Delta(\theta)})\oplus\mathsf{H}_4(k_R)\oplus\mathsf{H}_4(k_S)$,

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on July 10,2023 at 04:23:22 UTC from IEEE Xplore. Restrictions apply.

$c_1 = u_0^{r_1 \mathcal{Q}(\theta)} = u_0^{r_1 \prod_{i=\ell+2}^{\ell+k}(\theta+\rho_i)(\theta+\rho_{\ell+1})} = u^{r_1(\theta+\mathsf{H}_2(\mathsf{id}^*))}$, $c_2 = c_1^{\eta} = v^{r_1(\theta+\mathsf{H}_2(\mathsf{id}^*))}$, where $\Delta(\theta) = \frac{1}{\theta}(\prod_{i=\ell+2}^{\ell+k}(\theta+\rho_i) - \prod_{i=\ell+2}^{\ell+k}\rho_i)$. Note that if $\mathcal{Z} = e(g_0, u_0)^{r_1 \mathcal{P}(\theta)}$, $c_0 = M_b \cdot e(g,u)^{r_1} \oplus \mathsf{H}_4(k_R) \oplus \mathsf{H}_4(k_S)$. Finally, $\mathcal{B}$ produces a challenge ciphertext $\mathsf{ct}_{\mathsf{id}} = (c_0, c_1, c_2, \mathsf{R}_0, \mathsf{R}_1)$.

- **Guess**: $\mathcal{A}$ outputs a guess $\beta \in \{0,1\}$ and $\mathcal{B}$ gives the same guessed bit. In this proof, the advantage of discerning the well-formed authorized token from a random authorized token for $\mathcal{A}$ is negligible due to the fact that distinguishing part of well-formed authorized token $(s_1 = g_1^{-s}, s_2 = u^s \prod_{i=1}^{n}(\theta+\mathsf{H}_2(\mathsf{id}_j)), s_3 = \mathsf{H}_3(e(g,u)^s) \cdot u^t, s_4 = \mathsf{sk}^1_{\mathsf{id}_i} \cdot v^{-t})$ and part of random one $(s_1' = g_1^{-s'}, s_2' = u^{s'} \prod_{i=1}^{n}(\theta+\mathsf{H}_2(\mathsf{id}_j)), s_3' = \mathsf{H}_3(e(g,u)^{s'}) \cdot u^{t'})$ is negligible, which was proved in [19]. Therefore, if $\mathcal{A}$ with a non-negligible advantage breaks this security game, then the GDDHE assumption can be solved via the interaction with $\mathcal{A}$ with a non-negligible advantage, which completes the proof of this theorem. ■

Theorem 4: Assume that $\mathcal{A}$ can break the authenticity of our construction with non-negligible advantage, while making queries from the oracles EKGen, SKGen and the random oracle H_4. Then another algorithm called challenger $\mathcal{B}$ can also be constructed to solve the BDH problem with non-negligible advantage.

Proof: $\mathcal{B}$ is given a challenge BDH instance $(\mathsf{P}, \mathsf{P}^x, \mathsf{P}^y, \mathsf{P}^z)$. The goal of $\mathcal{A}$ is to compute $\mathcal{Z} = e(\mathsf{P},\mathsf{P})^{xyz}$. The interaction between $\mathcal{A}$ and $\mathcal{B}$ proceeds as follows:

- **Setup**: $\mathcal{B}$ produces and gives $\mathcal{A}$ the public parameter $\mathsf{pp} = (\mathbb{B}, \mathsf{H}_0, \mathsf{H}_1, \mathsf{H}_4, \mathsf{P}_0 = \mathsf{P}^x)$, where the oracles $\mathsf{H}_0, \mathsf{H}_1, \mathsf{H}_4$ are picked and controlled by $\mathcal{B}$.
- **Oracle Queries**: The following oracle queries are made and $\mathcal{B}$ performs the following steps to give the corresponding responses:
  - H_0 queries: If id_i has been in a tuple $(\mathsf{id}_i, T_i, \rho_i, r_i) \in \mathcal{L}_1$, then return T_i. Otherwise, randomly select $\rho_i \in \mathbb{Z}_p$ and a coin $r_i \in \{0,1\}$. If $r_i = 0$, set $T_i = \mathsf{P}^{\rho_i}$. Otherwise, set $T_i = \mathsf{P}^{z\rho_i}$. Finally, add $(\mathsf{id}_i, T_i, \rho_i, r_i)$ to $\mathcal{L}_1$ and respond T_i to $\mathcal{A}$.
  - H_1 queries: If id_i' has been in a tuple $(\mathsf{id}_i', T_i, \rho_i, r_i) \in \mathcal{L}_2$, then return T_i. Otherwise, randomly select $\rho_i \in \mathbb{Z}_p$ and a coin $r_i \in \{0,1\}$. If $r_i = 0$, set $T_i = \mathsf{P}^{\rho_i}$. Otherwise, set $T_i = \mathsf{P}^{z\rho_i}$. Finally, add $(\mathsf{id}_i', T_i, \rho_i, r_i)$ to $\mathcal{L}_2$ and respond T_i to $\mathcal{A}$.
  - H_4 queries: A list $\mathcal{L}$ is maintained to store the tuples (E_i, h_i) with the record of calls to H_4. If E_i is existed, h_i is then returned by $\mathcal{B}$. Otherwise, randomly pick a $h_i \in \mathbb{G}_1$ and add (E_i, h_i) to $\mathcal{L}$.
  - EKGen queries: Let id_i denote the input of the oracle EKGen. $\mathcal{B}$ can get $\mathsf{H}_1(\mathsf{id}_i) = T_i$, where $(\mathsf{id}_i, T_i, \rho_i, r_i)$ is the tuple in $\mathcal{L}_2$. If $r_i = 1$, $\mathcal{B}$ aborts. Otherwise, return $\mathsf{ek}_i = \mathsf{P}^{y\rho_i}$.
  - SKGen queries: Let id_i' denote the input of the oracle SKGen. $\mathcal{B}$ can get $\mathsf{H}_0(\mathsf{id}_i) = T_i$, where $(\mathsf{id}_i', T_i, \rho_i, r_i)$ is the tuple in $\mathcal{L}_1$. If $r_i = 1$, $\mathcal{B}$ aborts. Otherwise, return the partial secret key $\mathsf{sk}_{\mathsf{id}_i'} = (\mathsf{sk}^1_{\mathsf{id}_i'}, \mathsf{sk}^2_{\mathsf{id}_i'}, \mathsf{sk}^3_{\mathsf{id}_i'}) = (\mathsf{P}^{x\rho_i}, \mathsf{P}^{y\rho_i}, \mathsf{P}^{\rho_i})$.
- **Forgery**: $\mathcal{A}$ sends $(\mathsf{ct}, \mathsf{id}, \mathsf{id}^*)$. Let $\mathsf{id} = \mathsf{id}^*$. $\mathcal{B}$ performs the following actions:
  - Compute $\mathsf{H}_0(\mathsf{id}') = T$ and $\mathsf{H}_1(\mathsf{id}) = T'$. If the coins r_i, r_i' are not equal to 0 in both $(\mathsf{id}_i, T_i, \rho_i, r_i) \in \mathcal{L}_1$ and $(\mathsf{id}_i', T_i', \rho_i', r_i') \in \mathcal{L}_2$, then $\mathcal{B}$ aborts. otherwise, we can learn that $\mathsf{sk}^2_{\mathsf{id}_i'} = \mathsf{P}^{yzr}$ and $\mathsf{H}_0(\mathsf{id}) = \mathsf{P}^{xr'}$. Hence, $\mathsf{H}_4(k_{\mathcal{S}}) = \mathsf{H}_4(e(\mathsf{sk}^2_{\mathsf{id}_i'}, \mathsf{H}_1(\mathsf{id}))e(\mathsf{sk}^3_{\mathsf{id}_i'}, \mathsf{R}_0))$, where $e(\mathsf{sk}^2_{\mathsf{id}_i'}, \mathsf{H}_1(\mathsf{id})) = e(\mathsf{P}^{yzr}, \mathsf{P}^{xr'}) = \mathcal{Z}^{rr'}$ and $T = \mathsf{sk}^3_{\mathsf{id}_i'}$.
  - Parse partial ciphertext ct' as $(\mathsf{R}_0, \mathsf{R}_1, c_0)$. Compute $\zeta = 1/rr'$ and randomly take the tuple (E_i, h_i). Return $\mathcal{Z}' = (E_i \cdot e(T, \mathsf{R}_0))^{\zeta}$.

Please note that our simulation is perfectly done due to the fact that in the authenticity game the challenge $(\mathsf{ct}, \mathsf{id}, \mathsf{id}^*)$ satisfies $\mathsf{id} \notin \mathcal{O}_{\mathsf{SKGen}}$ and $\forall \mathsf{id}' \in \mathcal{O}_{\mathsf{EKGen}}, \mathsf{id}' \neq \mathsf{id}^*$. ■

B. Security Analysis

Theorem 5: If PDSM-FC is the IND-CPA, LR-CA and UNF-FA secure, then all the underlying attacks given in the threat model can be resisted in our constructed vehicle platoon system.

Proof: As proved in the **Theorems 2-5**, our vehicle platoon scheme can be deduced to be secure. The underlying attacks defined in the threat model are analyzed as below:

1) Impersonation Attacks: The attacks can be reduced as the attacks in the IND-CPA model. The adversarial goal is that the attackers can be any unauthorized users who have no decryption privilege by impersonating legitimate users to decipher the intended ciphertext. PDSM-FC permits a data owner to pick a random number to encode the data for an intended user, such that only the specified user who has an authorized private key can decode the data. Besides, PDSM-FC also allows a data user to transform the original ciphertext into a novel one for a set of intended users, such that only the user whose identity is within the authorization list can recover the data. Due to the fact that the secret key is produced according to its certified identity, non-authorization attackers having no corresponding private key fail to learn any data content.

2) Collusion Attacks: The attacks can be reduced as the attacks in the LR-CA model. Their purpose is to decrypt the valid ciphertext with an invalid decryption key although the attacker colludes with authorized users or cloud server to capture partial decryption keys. Note that if the decryption key is fully leaked to the adversaries, the attackers can directly recover the encrypted data. Then it is meaningless to measure the capabilities of attackers. In our work, two random seeds are chosen to encode the identity of each user. Hence, any new legitimate secret key via the conjunction of various partial private keys cannot be produced. That is to say, any malicious user cannot decipher the data content with a valid secret key produced via the collusion attacks.

3) Forgery Attacks: The attacks can be reduced as the attacks in the UNF-FA model. The goal is to forge a valid encryption key, thus forging a valid ciphertext and passing the verification. With our scheme, the attackers with invalid encryption keys fail to forge a valid encryption key since

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on July 10,2023 at 04:23:22 UTC from IEEE Xplore. Restrictions apply.

TABLE II
FUNCTIONALITY COMPARISONS OF OUR PDSM-FC WITH OTHER RELATED SCHEMES

Scheme	Cross-Domain Transformation	Data Authenticity	Stronger Security	Identity-based Setting
Cryptographic Signature [13]–[15]	✗	✓	✗	✗
Identity-based Signcryption Solution [16]	✗	✓	✗	✓
Other Signcryption Solution [17], [18]	✗	✓	✗	✗
Identity-based Proxy Re-encryption [23]–[25]	✗	✗	✗	✓
Other Proxy Re-encryption Solution [20]–[22], [26]–[32]	✗	✗	✗	✓
Identity-based Cross-domain Solution [34]	✓	✗	✗	✗
Other Cross-domain Solution [33], [35]	✓	✗	✗	✓
PDSM-FC	✓	✓	✓	✓

TABLE III
STORAGE COST COMPARISONS OF OUR PDSM-FC WITH RELATED SCHEMES

Scheme	Costs at client side			Costs at server side	
	pp storage	sk storage	at storage	Original ct storage	Transformed ct storage
MA [33]	$4\|\mathbb{G}_0\|+\|\mathbb{G}_1\|$	$2\|\mathbb{G}_0\|$	$3\|\mathbb{G}_0\|$	$3\|\mathbb{G}_0\|$	$2\|\mathbb{G}_0\|+\|\mathbb{G}_1\|$
XJW+ [26]	$(3m+2)\|\mathbb{G}_0\|+\|\mathbb{G}_1\|$	$\|\mathbb{G}_0\|$	$3\|\mathbb{G}_0\|+\|\mathbb{G}_1\|$	$3\|\mathbb{G}_0\|+\|\mathbb{G}_1\|$	$4\|\mathbb{G}_0\|+\|\mathbb{G}_1\|$
JNL+ [35]	$5\|\mathbb{G}_0\|+\|\mathbb{G}_1\|$	$\|\mathbb{G}_0\|+2\|\mathbb{Z}_p\|$	$4\|\mathbb{G}_0\|+\|\mathbb{G}_1\|$	$4\|\mathbb{G}_0\|+3\|\mathbb{G}_1\|$	$2\|\mathbb{G}_0\|+\|\mathbb{G}_1\|$
PDSM-FC	$(m+8)\|\mathbb{G}_0\|+\|\mathbb{G}_1\|$	$4\|\mathbb{G}_0\|$	$7\|\mathbb{G}_0\|$	$4\|\mathbb{G}_0\|+\|\mathbb{G}_1\|$	$4\|\mathbb{G}_0\|+\|\mathbb{G}_1\|$

every encryption key is created via the master secret key, which leads to the failure of valid ciphertext generation. Besides, the collision-resistant hash function is used to hinder any manipulation of the ciphertexts, which can also result in the failure of the ciphertext verification due to the collision-resistant hash function property. ■

VI. PERFORMANCE EVALUATION

In this section, we first qualitatively comapre the functionalities ouf our methodology with related works. Then we perform theoretical analysis and comparisons in terms of computation and communication cost. Finally, we conduct experiments to compare the computation cost of each algorithm in related works to validate the practicality of PDSM-FC.

A. Functionality Comparison

In TABLE II, we conclude the functionality comparisons between different categories of related works and our PDSM-FC in the aspect of cross-domain transformation, data authenticity, strong security, and identity-based setting. Cross-domain transformation enables a platoon user (i.e., PL^A) to share its data in a ciphertext format with another vehicle user (i.e., PL^B) who transforms the previous ciphertext format into another ciphertext format, such that only users within the authorization list can access it. Data authenticity implies that the shared data is protected from being forged, edited or replaced by malicious platoon users. Strong security means that various malicious attacks such as collusion attacks, forgery attacks and impersonation attacks, can be resisted with the proposed scheme to ensure their robustness. Correspondingly, weak security indicates only chosen plaintext attacks can be blocked in the given scheme. Identity-based setting implies relatively lower computation costs due to few pairing computations involved, which ensures the practicality of the designed scheme. These desirable features make the designed scheme more practical and appropriate for real-world applications. From TABLE II, we can learn that the works [13]–[15], [17], [18] only ensure data authenticity but fail to realize the rest of the other characteristics. The work [16] can achieve data authenticity in the identity-based setting and the works [20]–[32] are constructed in the identity-based setting. Besides, we can also observe that the works [33]–[35] enable cross-domain transformation and in which the works [33], [35] are designed in the identity-based setting. Our PDSM-FC can simultaneously enable cross-domain transformation, data authenticity, and strong security in the identity-based setting. Based on the above analysis, we can summarize that the listed functionalities can be partially achieved in prior works while only our PDSM-FC can achieve all the desired functionalities.

B. Theoretical Analysis

The comparisons of storage and computation costs are correspondingly summarized in TABLE III and TABLE IV. Here, the computation cost indicates the running time of each algorithm in PDSM-FC. For example, the computation cost of the decryption (Dec) algorithm refers to the time consumption to implement this algorithm. The communication

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on July 10,2023 at 04:23:22 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
COMPUTATION COST COMPARISONS OF OUR PDSM-FC WITH RELATED SCHEMES

Scheme	Costs at registry authority side		Costs at client side			Costs at server side
	Setup	SKGen	Encrypt	Authorize	Decrypt	Transform
MA [33]	$2e_0$	$4e_0$	$2p$	$2e_0$	$2p\|\|e_0 + p$	$2e_0 + p$
XJW+ [26]	$(3m+1)e_0 + p$	e_0	$4e_0 + p + e_1$	$(n+5)e_0 + e_1$	$ne_0 + e_1 + 2p\|\|ne_0 + e_1 + 2p$	$ne_0 + e_1 + 2p$
JNL+ [35]	e_0	$6e_0$	$4e_0 + 2p + 2e_1$	$6e_0 + e_1 + p$	$e_1\|\|e_1 + p$	$8e_0 + 3p$
PDSM-FC	$(m+5)e_0 + p$	$3e_0$	$6e_0 + 2p + e_1$	$(n+2)e_0 + e_1$	$4p\|\|ne_0 + 2p + e_1$	$3p$

overhead refers to the storage cost of each algorithm to produce related parameters. For instance, the communication cost of the encryption (Enc) algorithm refers to the storage cost for the final generated ciphertext. Commonly speaking, the smaller the computation and communication cost, the higher the performance. For the vehicle platoon applications, since each vehicle in the platoon may need to access the data to complete the merging task, it is vital to ensure the smaller computation and communication cost, especially in the encryption and decryption phases. In our experiment, we consider the most costly cryptographic calculations: bilinear maps, exponentiations in $\mathbb{G}_0$ and exponentiations in $\mathbb{G}_1$, which are denoted as p, e_0 and e_1 respectively. m, n denote the maximum numbers of system users and users within the authorization list. Let $|\mathbb{G}_0|$, $|\mathbb{G}_1|$ and $|\mathbb{Z}_p|$ denote the length of a randomness in $\mathbb{G}_0$, $\mathbb{G}_1$ and $\mathbb{Z}_p$. A and B denote the computation cost of the original ciphertext decryption and re-encryption ciphertext decryption in $\mathsf{A}\|\mathsf{B}$ shown in TABLE IV.

From TABLE III, we can observe that the storage cost of the public parameter (pp) in the work [26] and PDSM-FC increases with the maximum number of system users while that in the works [33], [35] is almost constant. The storage cost of secret key (sk), authorization token (ak), original ciphertext (ct) and transformed ciphertext in the works [26], [33], [35] and PDSM-FC is always constant. We can also learn that our original ct storage is slightly lower than that of [35] and slightly higher than that of others [26], [33]. The storage cost of pp, sk, ak and transformed ct in our PDSM-FC is relatively higher than that of other works [26], [33]. The reason leading to this mainly stems from data authenticity guarantee requiring the storage of more related parameters.

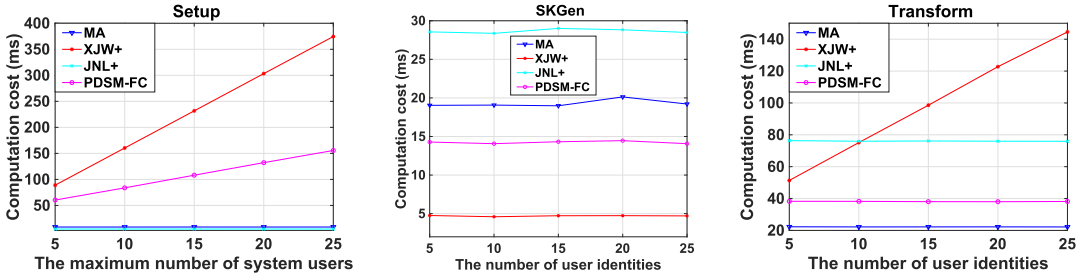
TABLE IV reveals that the computation cost of Setup in the work [26] and PDSM-FC grows with the maximum number of system users while it is always constant in other works [33], [35]. The computation cost of SKGen in our PDSM-FC is constant, smaller than that in [33], [35] but slightly higher than that in [26]. The computation cost of Encrypt in all the listed works is constant, and that in ours is slightly higher than that in others [26], [33], [35]. The computation cost of Authorize in the work [26] and our PDSM-FC follows a linear relationship with the number of authorized identities while that in others [33], [35] is constant. The computation cost of decrypting the original ciphertext in PDSM-FC is constant, slightly higher than that in [33], [35] while that in the work [26] increases linearly with the number of authorized identities [33], [35]. The computation cost of re-encrypted ciphertext decryption in PDSM-FC and the work [26] also linearly correlated with the number of authorized identities and is constant in other works [33], [35]. The computation cost of Transform in PDSM-FC is constant, slightly higher than that in [33], but much lower than that in the work [35], while that in the work [26] increases linearly with the number of authorized identities.

To summarize, our PDSM-FC enables constant sizes of secret key and ciphertext as well as constant numbers of encryption and decryption operations regardless of the shared vehicle identities. It only needs to perform one encryption to securely realize one-to-many data sharing with different platoon vehicles. This is much more efficient than the decryption-then-broadcasting solution which needs to repeatedly conduct multiple encryption operations under different public keys of various platoon vehicles. Therefore, PDSM-FC is more stable and scalable for real-world applications with arbitrary numbers of shared vehicles.

C. Experimental Analysis

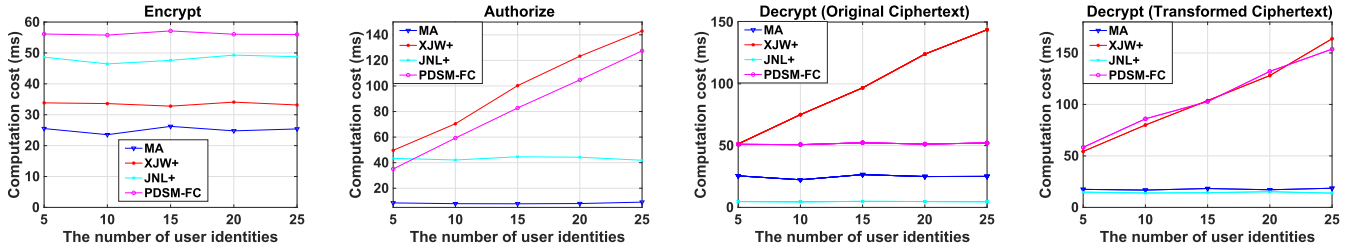
We run simulations to empirically compare our solution with prior works. We implement the cryptographic operations with Java. We adopt the Intellij IDEA-2018.2.5, Java 8 and the latest JPBC library [38] for simulations. All the experiments are conducted in a Lenovo server, which has 512GB SSD, 1TB mechanical hard disk, Intel(R) 8 Core(TM) i7-7820HK CPU@2.9 GHz and 16GB RAM. Besides, a supersingular elliptic curve denoted as $E(F_q) : z^2 = r^3 + r$ is used for our experimental simulations. Fig. 6 shows the comparisons of the computation cost at the registry authority and server side for different algorithms. From Fig. 6(a), we can obtain that the computation cost of the Setup algorithm in PDSM-FC and XJW+ [26] increases linearly with the number of system users while that in other works (MA [33] and JNL+ [35]) are almost stable regardless of the number of system users. The cost in PDSM-FC is much lower than that in XJW+, but slightly higher than that in MA and JNL+. From Fig. 6(b), we can conclude that the computation cost of Setup algorithm in all compared works are almost constant. The cost in PDSM-FC is much lower than that in MA and JNL+ but a bit higher than that in XJW+. From Fig. 6(c), it is easy to see that the computation cost of the Transform algorithm in XJW+ and JNL+ increases with the number of user identities while that

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on July 10,2023 at 04:23:22 UTC from IEEE Xplore. Restrictions apply.



(a) Setup algorithm (b) SKGen algorithm (c) Transform algorithm

Fig. 6. Comparison of costs at the registry authority and the server side for Setup, SKGen and Transform algorithms.



(a) Encrypt algorithm (b) Authorize algorithm (c) Decrypt algorithm (d) Decrypt algorithm

Fig. 7. Comparison of costs at the client side for Encrypt, Authorize and Decrypt algorithms.

in PDSM-FC and MA is almost constant irrespective of the number of user identities. The cost in PDSM-FC is much lower than XJW+ and JNL+ but slightly higher than MA.

Fig. 7 shows the comparisons of computation cost at the client side. From Fig. 7(a), it is intuitive to conclude that the computation cost of the Encrypt algorithm in all works keeps stable and is independent of the number of identities. The cost in our PDSM-FC is a bit higher than that in other works. This is because achieving data authenticity requires more computation operations. Specifically, the reason resulting in this inefficiency mainly stems from that the encryption key parts associated with an identity of an encryptor are used for secure signature generation in the encryption part, which directly leads to the relatively long-length ciphertext. As seen from Fig. 7(b), the computation cost of the Authorize algorithm in PDSM-FC and XJW+ follows a linear relationship with the number of user identities while that in MA and JNL+ is almost constant. We can learn from Fig. 7(c) that the computation cost of the Decrypt algorithm on the original ciphertext in XJW+ increases linearly with the number of user identities while that in JNL+, MA and PDSM-FC is almost stable. Besides, we can observe that the computation cost of the Decrypt algorithm on the original ciphertext in PDSM-FC is slightly higher than that in JNL+ and MA. From Fig. 7(d), we can conclude that the computation cost of the Decrypt algorithm on the transformed ciphertext in XJW+ and PDSM-FC grows with the number of user identities while that in MA and JNL+ is almost stable. Besides, we can know that the computation cost of the Decrypt algorithm on the transformed ciphertext in PDSM-FC is almost the same as that in XJW+ but a bit higher than that in JNL+ and MA. This is because decrypting long-length transformed ciphertext of course needs to consume lots of computing resources.

To summarize, the computation cost at the client side is relatively low (within 0.15s) in PDSM-FC, which makes it practical for the real-world applications.

VII. Conclusion and Future Works

In this paper, we propose PDSM-FC, a privacy-preserving data share mechanism with flexible cross-domain authorization for intelligent autonomous vehicle platoon. With our PDSM-FC, not only the authenticity of shared platoon data but also flexible cross-platoon authorization can be achieved without any exposure of the authorization token privacy and platoon data content. We give comprehensive security analysis and proofs to demonstrate the practicality of our PDSM-FC in resisting different types of attacks. Theoretical and experimental evaluations indicate the effectiveness and robustness of PDSM-FC in comparison with existing state-of-the-art solutions.

In the future, we will extend our PDSM-FC to achieve more stronger security against the chosen ciphertext attacks. Additionally, we are interested in exploring more practical functionalities based on our PDSM-FC, such as vehicle identity anonymity, dynamic membership updating, forward security, etc.

References

[1] S. Mumtaz, A. Al-Dulaimi, H. Gacanin, and A. Bo, "Block chain and big data-enabled intelligent vehicular communication," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 3904–3906, Jul. 2021.

[2] S. Mumtaz *et al.*, "Licensed and unlicensed spectrum for future 5G/B5G wireless networks," *IEEE Netw.*, vol. 33, no. 4, pp. 6–8, Jul./Aug. 2019.

[3] J. Sun, Y. Yuan, M. Tang, X. Cheng, X. Nie, and M. U. Aftab, "Privacy-preserving bilateral fine-grained access control for cloud-enabled industrial IoT healthcare," *IEEE Trans. Ind. Informat.*, early access, Dec. 13, 2021, doi: 10.1109/TII.2021.3133345.

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on July 10,2023 at 04:23:22 UTC from IEEE Xplore. Restrictions apply.

[4] M. Dikmen and C. Burns, "Trust in autonomous vehicles: The case of Tesla autopilot and summon," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 1093–1098.

[5] S. Yao, J. Zhang, Z. Hu, Y. Wang, and X. Zhou, "Autonomous-driving vehicle test technology based on virtual reality," *J. Eng.*, vol. 2018, no. 16, pp. 1768–1771, Nov. 2018.

[6] G. Rong *et al.*, "LGSVL simulator: A high fidelity simulator for autonomous driving," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.

[7] A. K. Sangaiah, J. S. Ramamoorthi, J. J. P. C. Rodrigues, M. A. Rahman, G. Muhammad, and M. Alrashoud, "LACCVoV: Linear adaptive congestion control with optimization of data dissemination model in vehicle-to-vehicle communication," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5319–5328, Aug. 2021.

[8] I. Al Ridhawi, M. Aloqaily, A. Boukerche, and Y. Jararweh, "Enabling intelligent IoCV services at the edge for 5G networks and beyond," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5190–5200, Aug. 2021.

[9] Y. Yang, Y. Xiao, and T. Li, "A survey of autonomous underwater vehicle formation: Performance, formation control, and communication capability," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 815–841, 2nd Quart., 2021.

[10] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal, "Platoon management with cooperative adaptive cruise control enabled by VANET," *Veh. Commun.*, vol. 2, no. 2, pp. 110–123, 2015.

[11] J. Sun, G. Xu, T. Zhang, H. Xiong, H. Li, and R. Deng, "Share your data carefree: An efficient, scalable and privacy-preserving data sharing service in cloud computing," *IEEE Trans. Cloud Comput.*, early access, Oct. 6, 2021, doi: 10.1109/TCC.2021.3117998.

[12] J. Sun, H. Xiong, S. Zhang, X. Liu, J. Yuan, and R. H. Deng, "A secure flexible and tampering-resistant data sharing system for vehicular social networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12938–12950, Nov. 2020.

[13] F. Zhang, R. N. Safavi, and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2004, pp. 277–290.

[14] Y. Li *et al.*, "SDABS: A flexible and efficient multi-authority hybrid attribute-based signature scheme in edge environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1892–1906, Mar. 2021.

[15] H. Xiong, Y. Bao, X. Nie, and Y. I. Asoor, "Server-aided attribute-based signature supporting expressive access structures for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1013–1023, Feb. 2020.

[16] L. Chen and J. L. Malone, "Improved identity-based signcryption," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2005, pp. 362–379.

[17] C. Hu, N. Zhang, H. Li, X. Cheng, and X. Liao, "Body area network security: A fuzzy attribute-based signcryption scheme," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 37–46, Sep. 2013.

[18] S. Belguith, N. Kaaniche, M. Hammoudeh, and T. Dargahi, "PROUD: Verifiable privacy-preserving outsourced attribute based signcryption supporting access policy update for cloud assisted IoT applications," *Future Gener. Comput. Syst.*, vol. 111, pp. 899–918, Oct. 2020.

[19] C. Delerablee, "Identity-based broadcast encryption with constant size ciphertexts and private keys," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, vol. 4843. Berlin, Germany: Springer, 2007, pp. 200–215.

[20] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 1998, pp. 127–144.

[21] B. Libert and D. Vergnaud, "Unidirectional chosen-ciphertext secure proxy re-encryption," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2008, pp. 360–379.

[22] H. Guo, Z. Zhang, J. Xu, N. An, and X. Lan, "Accountable proxy re-encryption for secure data sharing," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 1, pp. 145–159, Jan. 2021.

[23] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Berlin, Germany: Springer, 2007, pp. 288–306.

[24] C. K. Chu and W. G. Tzeng, "Identity-based proxy re-encryption without random oracles," in *Proc. Int. Conf. Inf. Secur.* Berlin, Germany: Springer, 2007, pp. 189–202.

[25] K. Liang, J. K. Liu, D. S. Wong, and W. Susilo, "An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing," in *Proc. Eur. Symp. Res. Comput. Secur.* Cham, Switzerland: Springer, 2014, pp. 257–272.

[26] P. Xu, T. Jiao, Q. Wu, W. Wang, and H. Jin, "Conditional identity-based broadcast proxy re-encryption and its application to cloud email," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 66–79, Jan. 2016.

[27] H. Deng, Z. Qin, Q. Wu, Z. Guan, and Y. Zhou, "Flexible attribute-based proxy re-encryption for efficient data sharing," *Inf. Sci.*, vol. 511, pp. 94–113, Feb. 2020.

[28] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds," *IEEE Trans. Dependable Secure Comput.*, early access, Apr. 29, 2021, doi: 10.1109/TDSC.2021.3076580.

[29] K. Liang *et al.*, "A DFA-based functional proxy re-encryption scheme for secure public cloud data sharing," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 10, pp. 1667–1680, Oct. 2014.

[30] X. Wang, X. Yang, C. Li, Y. Liu, and Y. Ding, "Improved functional proxy re-encryption schemes for secure cloud data sharing," *Comput. Sci. Inf. Syst.*, vol. 15, no. 3, pp. 585–614, 2018.

[31] F. Luo, S. Al-Kuwari, W. Susilo, and D. H. Duong, "Chosen-ciphertext secure homomorphic proxy re-encryption," *IEEE Trans. Cloud Comput.*, early access, Dec. 3, 2020, doi: 10.1109/TCC.2020.3042432.

[32] C. Ma, J. Li, and W. Ouyang, "A homomorphic proxy re-encryption from lattices," in *Proc. Int. Conf. Provable Secur.* Cham, Switzerland: Springer, 2016, pp. 353–372.

[33] T. Matsuo, "Proxy re-encryption systems for identity-based encryption," in *Proc. Int. Conf. Pairing-Based Cryptogr.* Berlin, Germany: Springer, 2007, pp. 247–267.

[34] T. Mizuno and H. Doi, "Hybrid proxy re-encryption scheme for attribute-based encryption," in *Proc. Int. Conf. Inf. Secur. Cryptol.* Berlin, Germany: Springer, 2009, pp. 288–302.

[35] P. Jiang, J. Ning, K. Liang, C. Dong, J. Chen, and Z. Cao, "Encryption switching service: Securely switch your encrypted data to another format," *IEEE Trans. Services Comput.*, vol. 14, no. 5, pp. 1357–1369, Sep. 2021, doi: 10.1109/TSC.2018.2876849.

[36] H. Deng *et al.*, "Identity-based encryption transformation for flexible sharing of encrypted data in public cloud," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3168–3180, 2020.

[37] S. Xu *et al.*, "Match in my way: Fine-grained bilateral access control for secure cloud-fog computing," *IEEE Trans. Dependable Secure Comput.*, early access, Jun. 11, 2020, doi: 10.1109/TDSC.2020.3001557.

[38] B. Lynn. *The Stanford Pairing Based Crypto Library*. Accessed: Dec. 2021. [Online]. Available: http://crypto.stanford.edu/jpbc/

[39] H. Hartenstein and L. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 6, pp. 164–171, Jun. 2008.

[40] S. Wen, G. Guo, B. Chen, and X. Gao, "Event-triggered cooperative control of vehicle platoons in vehicular ad hoc networks," *Inf. Sci.*, vol. 459, pp. 341–353, Aug. 2018.

[41] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 263–284, 1st Quart., 2016.

[42] X. Zhu, S. Jiang, L. Wang, and H. Li, "Efficient privacy-preserving authentication for vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 907–919, Feb. 2014.

[43] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2681–2691, Aug. 2015.

[44] X. Lin, X. Sun, X. Wang, C. Zhang, P.-H. Ho, and X. Shen, "TSVC: Timed efficient and secure vehicular communications with privacy preserving," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4987–4998, Dec. 2008.

[45] I. Ali, Y. Chen, N. Ullah, R. Kumar, and W. He, "An efficient and provably secure ECC-based conditional privacy-preserving authentication for vehicle-to-vehicle communication in VANETs," *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1278–1291, Feb. 2021.

[46] Y. Zhao, Y. Wang, X. Cheng, H. Chen, H. Yu, and Y. Ren, "RFAP: A revocable fine-grained access control mechanism for autonomous vehicle platoon," *IEEE Trans. Intell. Transp. Syst.*, early access, Aug. 26, 2021, doi: 10.1109/TITS.2021.3105458.

[47] B. Li, R. Liang, D. Zhu, W. Chen, and Q. Lin, "Blockchain-based trust management model for location privacy preserving in VANET," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3765–3775, Jun. 2021.

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on July 10,2023 at 04:23:22 UTC from IEEE Xplore. Restrictions apply.

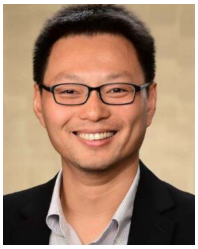
Jianfei Sun received the Ph.D. degree from the University of Electronic Science and Technology of China (UESTC). He is currently a Research Fellow at the School of Computer Science and Engineering, Nanyang Technological University. His research interests include network security and the IoT security.

Xiaochun Cheng received the Ph.D. degree in computer science from Jilin University in 1996. He has been a Computer Science EU Project Coordinator at Middlesex University London since 2012. He is currently with the Department of Computer Science, Middlesex University London. His research interest includes information security.

Guowen Xu received the Ph.D. degree from the University of Electronic Science and Technology of China in 2020. He is currently a Research Fellow with Nanyang Technological University, Singapore. His research interests include applied cryptography and privacy-preserving issues in deep learning.

Xingshuo Han is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include safety and privacy of deep learning, safety and security of autonomous vehicles, and intelligent transportation systems.

Tianwei Zhang (Member, IEEE) received the bachelor's degree from Peking University in 2011 and the Ph.D. degree from Princeton University in 2017. He is currently an Assistant Professor with the School of Computer Science and Engineering, Nanyang Technological University. His research focuses on computer system security. He is particularly interested in security threats and defenses in machine learning systems, autonomous systems, computer architecture, and distributed systems.



Mingjian Tang received the Ph.D. degree (Hons.) in computer science from La Trobe University, Melbourne, Australia, in 2009. He is currently with Westpac Banking Corporation, Sydney, NSW. He has participated in several industry-based research projects, including unsupervised fraud detection, unstructured threat intelligence, cyber risk analysis and quantification, and big data analytics.

Authorized licensed use limited to: Nanyang Technological University Library. Downloaded on July 10,2023 at 04:23:22 UTC from IEEE Xplore. Restrictions apply.