Anonymous and Efficient Authentication Scheme for Privacy-Preserving Distributed Learning

Yili Jiang, *Student Member, IEEE*, Kuan Zhang, *Member, IEEE*, Yi Qian, *Fellow, IEEE*, and Liang Zhou, *Senior Member, IEEE*

***Abstract*— Distributed learning is proposed as a promising technique to reduce heavy data transmissions in centralized machine learning. By allowing the participants training the model locally, raw data is unnecessarily uploaded to the centralized cloud server, reducing the risks of privacy leakage as well. However, the existing studies have shown that an adversary is able to derive the raw data by analyzing the obtained machine learning models. To tackle this challenge, the state-of-the-art solutions mainly depend on differential privacy and encryption techniques (e.g., homomorphic encryption). Whereas, differential privacy degrades data utility and leads to inaccurate learning, while encryption based approaches are not effective to all machine learning algorithms due to the limited operations and excessive computation cost. In this work, we propose a novel scheme to resolve the privacy issues from the anonymous authentication approach. Different from the two types of existing solutions, this approach is generalized to all machine learning algorithms without reducing data utility, while guaranteeing privacy preservation. In addition, it can be integrated with detection schemes against data poisoning attacks and free-rider attacks, being more practical for distributed learning. To this end, we first design a pairing-based certificateless signature scheme. Based on the signature scheme, we further propose an anonymous and efficient authentication protocol which supports dynamic batch verification. The proposed protocol guarantees the desired security properties while being computationally efficient. Formal security proof and analysis have been provided to demonstrate the achieved security properties, including confidentiality, anonymity, mutual authentication, unlinkability, unforgeability, forward security, backward security, and non-repudiation. In addition, the performance analysis reveals that our proposed protocol significantly reduces the time consumption in batch verification, achieving high computational efficiency.**

***Index Terms*— Distributed learning, privacy preservation, anonymous authentication, efficiency.**

Manuscript received November 29, 2021; revised March 20, 2022 and May 22, 2022; accepted May 26, 2022. Date of publication June 9, 2022; date of current version June 24, 2022. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. George Theodorakopoulos. *(Corresponding author: Yi Qian.)*

Yili Jiang is with the College of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China, and also with the Department of Electrical and Computer Engineering, University of Nebraska–Lincoln, Omaha, NE 68106 USA (e-mail: yilijiang@huskers.unl.edu).

Kuan Zhang and Yi Qian are with the Department of Electrical and Computer Engineering, University of Nebraska–Lincoln, Omaha, NE 68106 USA (e-mail: kuan.zhang@unl.edu; yi.qian@unl.edu).

Liang Zhou is with the College of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: liang.zhou@njupt.edu.cn).

Digital Object Identifier 10.1109/TIFS.2022.3181848

I. Introduction

THE integration of machine learning (ML) and cloud computing has been developed rapidly in the past decade. By collecting data from the Internet of Things (IoT) devices, cloud server trains ML models to provide intelligent services for diverse applications [1]–[3]. However, with the implementation of large-scale and ubiquitous IoT devices, tremendous amount of data has been generated and transmitted to cloud server, leading to significant communication and computation overheads. To solve this problem, distributed learning (DL) is proposed to execute ML in a collaborative manner. Specifically, instead of uploading raw data, the participated devices train the ML model locally and simply send their model parameters to the cloud server. The cloud server then aggregates into a global ML model. This approach has attracted great attention from both academia and industry. For instance, Google's keyboard team proposes a distributed approach to train a recurrent neural network for next-word prediction with 1.5 million participants [4]. Moreover, since the participants are unnecessary to upload their data to the cloud server, the risks of privacy leakage are reduced in DL systems.

Despite the appealing benefits, DL is still subject to privacy violations when the cloud server is not fully trusted. He *et al.* [5] implemented three attacks (white-box attack, black-box attach, and query-free attack) in distributed scenarios, and show that the cloud is able to recover the input of the ML model with the knowledge of full/partial model parameters. Once the raw data is recovered, the personal information of participants is disclosed. Thus, protecting the model parameters from untrusted cloud server is necessary. However, since the cloud server plays a role of data aggregator, it is required to process the model parameters meanwhile. Consequently, at the cloud server side, achieving secure parameter aggregation without revealing the raw parameters is a challenge in DL.

To tackle this challenge, deploying encryption techniques or differential privacy (DP) are two primary approaches in the state-of-the-art solutions. 1). Encryption techniques, such as homomorphic encryption, secure multiparty computation, and secret sharing, are employed to design secure data exchange/aggregation protocols. Benefiting from the protocols, the cloud is able to calculate the global model without accessing the model parameters of individual participant. However, this approach is not generalized for all ML algorithms in DL, since it may only support a limited number of operations

over addition and multiplication in practice. Additionally, it consumes excessive communication/computation resources due to heavily cryptographic primitives [6]. 2). DP is deployed to obfuscate information by introducing random noise to the raw data or model parameters. Although DP-based schemes are generalized for all ML algorithms in DL, the randomness reduces data utility, bringing undesired accuracy degradation to ML models and longer latency in model convergence. Hence, DP is ineffective for accuracy-sensitive situations. Because of the limitations of the existing solutions, it is urgent to develop a new computationally efficient solution that is generalized for all ML algorithms without data utility reducing.

In this paper, we propose a novel scheme with anonymous authentication to address the above challenges for privacy-preserving distributed learning. Specifically, the proposed scheme is based on the fact that if the participants in the DL system are anonymous, then although the raw data is recovered as discussed in [5], the adversary cannot link the data to the corresponding participant's real identity, achieving the privacy preservation. Different from the two existing approaches, the advantages of our new scheme are four folds:

1) The anonymity protects a specific participant's privacy since its real identity is confidential. Besides anonymity, various security properties are also achieved, such as confidentiality, mutual authentication, unlinkability, unforgeability, forward security, backward security, non-repudiation, and so on. The detailed definitions of these security properties are described in section III-B.
2) Compared with DP-based approach, our approach avoids the degradation of data utility caused by the introduced random noise.
3) Compared with the encryption-based approach, our approach is more generalized and effective for all ML algorithms, being more suitable for DL systems.
4) Our scheme can be integrated with the detection schemes against data poisoning attacks and free-rider attacks. In addition to privacy violations, DL is vulnerable to attacks from internal attackers who may upload forged data to poison the global ML model (data poisoning attacks) or upload meaningless data to obtain the global ML model "for free" (free-rider attacks) [7]–[9]. Since the detection schemes are performed over the plaintext, they are invalid in the existing encryption/DP-based schemes. Therefore, by integrating with the detection schemes, our scheme provides more potential benefits in security and privacy compared with the two existing approaches.

As an important research direction, several anonymous authentication schemes have been proposed in some applications. The most related studies to this paper are the authentications schemes in e-health systems or vehicular ad hoc networks (VANET) because they have very similar network features. In e-health, most of the authentication schemes achieve various security properties based on the public key cryptography (PKC). However, typical PKC-based schemes require a certification authority (CA) to issue and maintain the certificates of users' public keys, consuming extra storage and computation resources [10]. Moreover, verifying the certificates leads to more computation overhead and longer latency during the authentication procedure. Although some certificateless authentication schemes are proposed to eliminate CA and improve efficiency [11], [12], those schemes still suffer from time inefficiency in large-scale and time-sensitive scenarios (e.g. VANET and DL). In VANET, to address the time efficiency, some batch verification supported authentication schemes are proposed. Whereas, these schemes may highly depend on a tamper-proof device (TPD), which is a strong assumption in practice, or may sacrifice some security properties [13]. Therefore, existing authentication schemes in VANET and e-health systems cannot be deployed in DL directly. We are necessary to design an anonymous authentication scheme that is more time-efficient and independent of the strong assumptions in practice (e.g., TPD), such that the scheme is more suitable for DL.

In this paper, we design a pairing-based certificateless signature (PCLS) scheme and propose an anonymous authentication protocol based on the designed signature scheme for privacy-preserving DL. Compared with the existing research work, the proposed protocol supports batch verification and various security properties, being more computational efficient and secure. Therefore, it is more suitable for DL systems. The main contributions of this paper are summarized as follows:

- We propose a novel scheme to resolve the privacy issues in DL by deploying anonymous authentication. Compared with the existing approaches, our new scheme is effective for all ML algorithms without reducing data utility, being more generalized for DL systems. In addition, our scheme has potentials to be integrated with the defense schemes to detect data poisoning attacks and free-rider attacks.
- We design a new generalized certificateless signature scheme PCLS, which is built on pairing-based cryptography. Based on the designed signature scheme, we propose an anonymous and efficient authentication protocol which supports batch verification (AAPBV). Different from the batch verification in VANET, our proposed scheme is independent of TPD.
- We design an algorithm of dynamic batch verification (DBV) which can dynamically adjust the batch size during batch verification. In conventional batch verification, the batch size is not adjustable. All the messages are discarded if the batch verification fails. In our scheme, when the failure occurs, DBV is launched to adjust the batch size and redo verification. Benefiting from DBV, more legitimate messages are saved and less messages are discarded. Consequently, more valid data contributes to model learning, improving the convergence speed.
- We provide formal proof and analysis about the achieved security properties of the proposed protocol. In addition to anonymity, the proposed protocol satisfies confidentiality, unlinkability, unforgeability, non-repudiation, mutual authentication, forward security, and backward security. Among the eight security properties, the first five features are realized based on the designed PCLS while other three features are realized based on the proposed AAPBV.

- We conduct extensive simulations to evaluate our proposed protocol in terms of computational efficiency. The performance evaluation indicates that the proposed protocol can significantly reduce the computational time of authentication. Especially with batch verification, time consumption is reduced by more than 70% comparing with [14], [15].

The rest of this paper is organized as follows. Section II introduces the related work. Section III provides the system model and design goals. Section IV describes the proposed certificateless signature scheme, which is the design basis of our proposed protocol. After that, Section V presents the details of the proposed protocol. Section VI provides formal proof and detailed analysis of the achieved security properties. Section VII discusses simulation results and Section VIII provides conclusions.

II. Related Work

To prevent an adversary from deriving private information from the model parameters, the state-of-the-art solutions in DL systems mainly rely on DP and encryption techniques. In this section, we first discuss the related work in these two directions. Since our proposed scheme is to solve the privacy issues from the anonymous authentication, which is different from the two existing directions, we then discuss the related work of authentication schemes from the other application scenarios. As VANET and e-health systems have similar network features with a DL system, we focus on the related work of authentication schemes in these two scenarios.

A. DP

DP has been proposed as a powerful tool to protect the original information in data analysis and data publishing [16], [17]. It obfuscates the raw data by introducing randomized noise [18]. Zhang *et al.* [19] proposed to dynamically add noise data into the updated model parameters, such that the attackers are unable to recover the accurate model parameters. Consequently, the attackers cannot further derive the participants' personal data from the model parameters. In addition, DP is also employed to add noise into the participants' raw data before training the ML model [20], [21]. In this way, the attackers can only obtain the obfuscated data even in the best case. Although DP-based approach has significant efficiency in data transmissions, it degrades data utility [22]. The lower data utility leads to inaccurate ML model [23].

B. Encryption Techniques

Homomorphic encryption, secure multiparty computation, and secret sharing are three popular encryption techniques employed in privacy-preserving DL. Phong *et al.* [24] deployed additively homomorphic encryption into a collaborative learning system where the server can aggregate the model parameters over ciphertexts. Mohassel *et al.* [25] designed a protocol based on secure multiparty computation to implement privacy-preserving ML. Jia *et al.* [26] proposed a secret sharing based protocol in hierarchical distributed learning. Based on the protocol, the cloud server can obtain the parameter summation without accessing to the individual parameters. These techniques can be combined to achieve enhanced security and privacy [27]. Although the encryption based approaches can guarantee that the attackers cannot derive the personal data of participants, they may not support all ML algorithms in practice. Since only limited number of operations over addition and multiplication can be executed, not all ML algorithms can converge in this case.

C. Authentication

Since the traditional PKC-based authentication systems have high computation and storage overheads caused by maintaining the certificates of public keys, certificateless authentication has been proposed as a promising approach. Liu *et al.* [11] designed a certificateless signature scheme and proposed an anonymous authentication protocol based on the designed scheme. The proposed protocol achieves various security properties. Zhang *et al.* proposed a light-weight protocol [12] which enhances the security of [11]. However, both the works [11], [12] are not suitable for large-scale applications, since authenticating large-scale messages results in long latency and high computation overhead. To solve this problem, batch verification supported authentication protocols are proposed [14], [15], even though they may sacrifice some security properties or depend on some strong security assumptions (e.g. TPD) [28]. As the traditional batch verification generally has a fixed batch size or random batch size [29], all the messages are discarded once the batch verification fails. To reduce the number of the discarded messages, dynamic batch verification is proposed to adjust the batch size [30].

In this work, we propose a new scheme for privacy-preserving DL by employing anonymous authentication. Particularly, we design a pairing-based certificateless signature scheme, and propose an anonymous authentication protocol based on the designed signature scheme. To improve the computational efficiency, the proposed protocol is designed to support batch verification. In addition, we design an algorithm of DBV to reduce the number of discarded messages during batch verification.

III. System Model and Design Goals

In this section, we introduce the system model, security model, design goals, and preliminaries.

A. System Overview and Security Model

Fig. 1 illustrates the involved entities and workflow of a DL system, including N distributed participants and a centralized cloud server. The detailed descriptions of entities and workflow are presented as follows.

- *Participants*: Participants refer to individuals that are equipped with intelligent devices, such as laptop, smart phone, tablet, and so on, which have sufficient communication, computation, and storage capabilities. These devices collect the participants' information (e.g., browsing history, shopping habits, screen time, etc.) and generate a unique dataset. Other equipment may also contribute

to the dataset. For instance, a smart watch can count a participant's daily steps; wearable sensors can collect the health status information. The dataset is stored in the device locally and updated in real time. In DL, each participant uses the individual dataset to train a ML model locally and uploads the model parameters to the cloud server. Since different participant maintains a different dataset, the obtained model parameters can be varying.

- *Cloud Server*: The cloud server is also know as the parameter server. It receives the model parameters from participants and aggregates into a global ML model. The global ML model is then transmitted to all participants for next-iteration training. With sufficient iterations and participants, the global ML model can be corrected accurately. Since a global ML model may converge after hundreds or even thousands rounds, the aggregation procedures should be time-efficient in each round. For simplification of discussion, we may use "cloud" to denote "cloud server" and "global model" to denote "global ML model" in this paper.
- *Trusted Authority (TA)*: TA is a powerful system manager that is responsible for system initialization, participant registrations, key generation, identity management, and so on. In this work, when there is a potential malicious participant who uploads false data to damage the global model, TA is able to identify the real identity of the participant. After system initialization and participant registrations, TA can be off until reactivated by a request of identifying a malicious participant.

Considering N participants and a cloud server, the complete workflow of DL consists of the following steps.

Step 1: The cloud server initializes a global model (e.g., set the model parameters as all zeros) and distributes the model to all participants.

Step 2: After receiving the global model, each participant continues to train the model on its local dataset and obtains the updated model parameters W_n. $W_n(\forall n \in [1, N])$ is then uploaded to the cloud server.

Step 3: The cloud server aggregates the model parameters (e.g., $W_g = \frac{1}{N}\sum_{n=1}^{N} W_n$). The global ML model is corrected with W_g and sent to the participants for next-round training.

Step 4: Repeat Steps 2, 3 until the global model converges.

In this paper, we mainly focus on steps 2, 3 in a single iteration and consider the anonymous authentication between the participants and the cloud server. On the one hand, the cloud server can verify the legitimate participants and take their updated parameters into aggregation. On the other hand, although the cloud server may recover some raw data via W_n, it is unable to reveal the identity of the corresponding participant.

The security model is described as follows.

- The cloud server is honest-but-curious. It means that the cloud server can honestly follow the protocols to authenticate participants, aggregate the model parameters, and generate the global model. However, the cloud server is curious about the private data of the participants. It may attempt to recover a participant's raw data by analyzing the received model parameters and reveal the corresponding real identity. Therefore, the cloud server is possibly to be an attacker that threatens the privacy of the participants.
- A participant can be malicious. Particularly, 1) a legal participant can lunch data poisoning attacks and free-riding attacks by uploading false model parameters. It is noted that these two attacks can be detected by integrating the existing detection schemes [8], [9] into the proposed scheme. 2) An illegal participant may forge a valid signature to sign a false message such that the false message can be authenticated by the cloud successfully. 3) A participant is curious about the transmissions between the cloud and another participant. It may attempt to reveal the original data in the transmissions and link the data to a specific participant. Thus, the other participants' privacy can be threatened by a malicious participant.
- The system may be attacked by man-in-the-middle (MITM) attack, where an attacker stays between the cloud and a participant. The attacker relays and alters the communications between the cloud and a participant to make the two parties believe that they are directly communicating with each other.
- TA is fully trusted. It always provides reliable service during the system initialization, participant registrations, and key generation. It stores the real identities of the participants and keeps them confidential. When a participant is detected as malicious, TA is able to reveal its real identity to the cloud server.

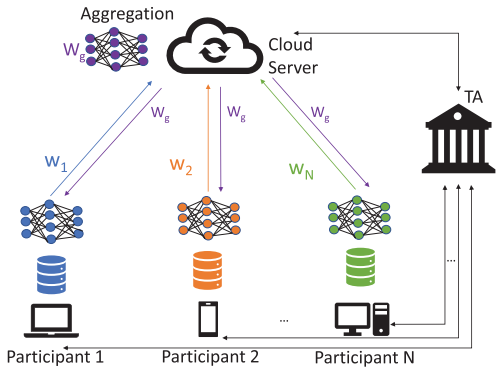


Fig. 1. System model.

B. Design Goals

Under the security model, we have the following design goals in this paper.

- *Confidentiality:* Confidentiality protects the attackers from revealing the participants' sensitive information. In other words, any entities other than the cloud cannot obtain the model parameters of participants.
- *Anonymity:* Any entities including the cloud (except TA) cannot derive the real identities of the participants in the system. In other words, the participants' real identities are confidential to any entities other than TA.
- *Mutual Authentication:* The cloud and the participants can authenticate each other to verify the sources and

TABLE I
NOTATION DEFINITIONS

Variable	Definition
(X', X'')	full private key
(Y', Y'')	full public key
$\sigma = (k, U)$	signature
W_n	plaintext of model parameters
$W_n^\star$	ciphertext of model parameters
W_g	global model parameters
Enc_s	function of symmetric key encryption
Dec_s	function of symmetric key decryption
S_n^j	session key of session j
$\mathbb{G}_1$	cyclic additive group
$\mathbb{G}_2$	cyclic multiplicative group
$\mathbb{Z}_p^\star$	$\{1, 2, 3, ..., p-1\}$
H_1, H_2, H_3	hash functions
$\{0,1\}^\star$	a sequence of binary number
t_e	time of one exponentiation in $\mathbb{G}_2$
t_m	time of one multiplication in $\mathbb{G}_1$
t_p	time of one pairing

destinations of the messages. Specifically, a mutual authentication between two entities can guarantee that the message is from the claimed source and reaches the intended destination. In addition, the system is expected to be secure from MITM attack.

- *Unlinkability:* The transmissions of any two messages should not be linked to the same participants. It means that an attacker cannot verify if two messages are from the same participant.
- *Unforgeability:* An attacker in the system cannot forge a valid signature to sign messages, which can be successfully authenticated by the cloud. Even the TA should not be able to forge any valid signatures.
- *Forward Security:* If the private key of a participant is compromised, the previous session keys between it and the cloud remain secure. Particularly, even though the current session key/private key is disclosed, the previous transmitted data between the cloud and the participant cannot be revealed.
- *Backward Security:* If the private key of a participant is compromised, the future session keys between it and the cloud cannot be revealed. In other words, even thought the current session key/private key is disclosed, the future communications between the cloud and the participant remain secure.
- *Non-repudiation:* When a participant updates the model parameters to the cloud, once the request is authenticated, the participant cannot deny that it has contributed to the global model. In this way, if a participant is detected that it uploaded false data to the cloud, it cannot deny its behavior.
- *Computational Efficiency:* The authentication protocol should be computationally efficient. Especially, when a large-scale DL system involves plenty of participants, the cloud can authentication them with high time efficiency.

C. Preliminaries

In this part, we review the Bilinear Pairing and the hard problem assumptions which are used to design our proposed scheme and prove the security.

1) Bilinear Pairing: Let $\mathbb{G}_1$ be a cyclic additive group with a prime order p. $\mathbb{G}_2$ be a cyclic multiplicative group with the same prime order p. A mapping $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is an admissible bilinear pairing that satisfies the following properties.

1) Bilinearity: $e(aV, bQ) = e(V, Q)^{ab}$, given $V, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p^\star$. This can be further expressed as $e(V + R, Q) = e(V, Q)e(R, Q)$ and $e(V, Q + R) = e(V, Q)e(V, R)$, $\forall V, Q, R \in \mathbb{G}_1$.
2) Non-degeneracy: $e(V, Q) \neq 1_{\mathbb{G}_2}$, where $V, Q \neq 1_{\mathbb{G}_1}$. $1_{\mathbb{G}_1}, 1_{\mathbb{G}_2}$ denote the identity element of group $\mathbb{G}_1, \mathbb{G}_2$ respectively.
3) Symmetry: $e(V, Q) = e(Q, V)$.
4) Computability: e is efficiently computable.

To prove the security of our scheme, we consider the following hard problem assumptions in $\mathbb{G}_1$.

2) Decisional Diffie–Hellman Problem (DDHP) Assumption: Considering a cyclic group $\mathbb{G}_1$ of the prime order p, for the tuple (aQ, bQ, V), where $Q, V \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p^\star$, the advantage for any polynomial probabilistic time adversary to determine whether $V = abQ$ holds is negligible.

3) Computational Diffie–Hellman Problem (CDHP) Assumption: Considering a cyclic group $\mathbb{G}_1$ of the prime order p, for the tuple (Q, aQ, bQ), where $Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p^\star$, it is intractable to compute abQ within polynomial time.

4) Discrete Logarithm Problem (DLP) Assumption: Given a cyclic group $\mathbb{G}_1$ of the prime order p and $V, Q \in \mathbb{G}_1$, it is intractable within polynomial time to find an integer $n \in \mathbb{Z}_p^\star$, such that $V = nQ$ whenever n exists.

IV. DESIGN BASIS: PCLS

In this section, we propose a pairing-based certificateless signature scheme, PCLS, which lays down a design basis for the proposed anonymous authentication protocol. In the proposed PCLS, there are two main modifications compared with the typical certificateless signature scheme. First, instead of setting the signer's real identity as the partial public key, we use the hash value of the signer's real identity as the partial public key, protecting the real identities from disclosure. Second, we modified the signature generation and message verification to support batch verification, such that the authentication is more efficient. The PCLS scheme involves the following steps. Definitions of the notations used in this section can be found in Table I.

1) Setup: Let $\mathbb{G}_1$ denote a cyclic additive group of a prime order p. $\mathbb{G}_2$ denotes a cyclic multiplicative group with the same prime order p. p satisfies $p > 2^l$, where l is the security parameter. Consider P is a generator of $\mathbb{G}_1$ and $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is a bilinear pairing operator. TA randomly selects $X_{TA} \in \mathbb{Z}_p^\star$ as its private key and computes $Y_{TA} = X_{TA}P$ as its public key. Then TA picks two hash functions H_1 and H_2, where $H_1 : \{0,1\}^\star \to \mathbb{G}_1$, $H_2 : \{0,1\}^\star \times \mathbb{G}_1 \to \mathbb{Z}_p^\star$. After that, TA publishes $(l, p, P, e, \mathbb{G}_1, \mathbb{G}_2, H_1, H_2, Y_{TA})$ as the system parameters. The private key X_{TA} is kept by TA and remains as a secret.

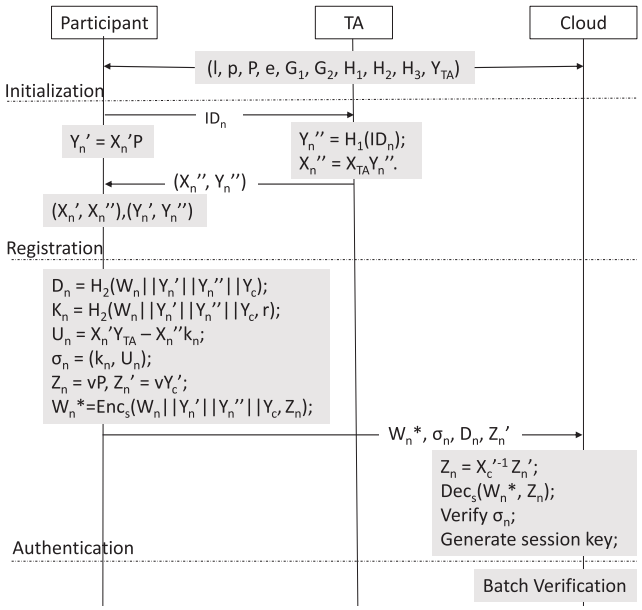


Fig. 2. Proposed protocol.

2) Key-Extract: The signer randomly selects $X' \in \mathbb{Z}_p^\star$ as its partial private key and calculates $Y' = X'P$ as its partial public key. The signer computes $Y'' = H_1(ID)$ as the other partial public key and sends Y'' to TA, where ID is its true identity. Given Y'', TA then sets $X'' = X_{TA}Y''$ and sends to the signer through a secure channel. The signer uses (X', X'') as its full private key and (Y', Y'') as its full public key.
3) Sign: Before sending a message m, the signer randomly chooses $r \in \mathbb{Z}_p^\star$ and signs the message as follows:

$$k = H_2(m||Y'||Y''||Y_{TA}, rP), \tag{1}$$

$$U = X'Y_{TA} - X''k. \tag{2}$$

The signature is $\sigma = (k, U)$ on message m.
4) Verify: After receiving the message m with signature σ, the verifier checks

$$e(Y', Y_{TA}) \stackrel{?}{=} e(U, P)e(Y'', Y_{TA})^k. \tag{3}$$

The message is accepted if the checking holds.

V. The Proposed Authentication Protocol

In this part, we design an anonymous authentication protocol AAPBV based on the proposed PCLS. The proposed AAPBV effectively achieves computational efficiency and various security properties between the cloud and the participants in DL. Firstly, we give an overview of the proposed protocol. Then we describe the protocol in details.

A. Overview of the Proposed Protocol

As shown in Fig. 2, the proposed AAPBV involves four stages, including initialization, registration, authentication, and batch verification. Particularly, in the initialization stage, the TA initializes the system parameters and publishes to the participants and the cloud. In the registration stage, a participant registers to TA with its real identity and generates its full public key and private key assisted by TA. In the authentication stage, a participant generates a valid signature, signs the message, and uploads to the cloud. The cloud then authenticates the participant and decrypts the message. To improve the computational efficiency in large-scale DL, the cloud can authenticate a batch of messages at the same time. In the batch verification stage, the batch verification may fail if there are illegal participants who upload messages signed by invalid signatures. In this case, the cloud launches the DBV algorithm to dynamically shrink the batch size and redo verification. Thus, the illegitimate messages may be filtered. In this way, the number of discarded messages is reduced and more legal messages are saved for machine learning.

B. Details of the Proposed Protocol

We show the details of the proposed protocol as follows.

1) *Initialization:* Given the security parameter l, TA initializes the system and generates the system parameters. Let $\mathbb{G}_1$ denote a cyclic additive group of a prime order $p > 2^l$. $\mathbb{G}_2$ denotes a cyclic multiplicative group with the same prime order p. P is a generator of $\mathbb{G}_1$ and $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is a bilinear pairing operator. TA picks three secure hash functions H_1, H_2, and H_3, where $H_1 : \{0,1\}^\star \to \mathbb{G}_1$, $H_2 : \{0,1\}^\star \times \mathbb{G}_1 \to \mathbb{Z}_p^\star$, $H_3 : \{0,1\}^\star \to \mathbb{Z}_p^\star$. In addition, TA randomly selects $X_{TA} \in \mathbb{Z}_p^\star$ as its private key and computes $Y_{TA} = X_{TA}P$ as its public key. The private key X_{TA} is kept by TA and remains a secret. $(l, p, P, e, \mathbb{G}_1, \mathbb{G}_2, H_1, H_2, H_3, Y_{TA})$ is published as the system parameters.
2) *Registration:* Both participants and the cloud need register to the TA with their real identities before joining the system. For the cloud, firstly, it randomly selects $X_c' \in \mathbb{Z}_p^\star$ as its partial private key and calculates $Y_c' = X_c'P$ as its partial public key. Secondly, it sends its identity ID_c to the TA via a secure channel. TA then computes $Y_c'' = H_1(ID)$, $X_c'' = X_{TA}Y_c''$ and returns key pair (X_c'', Y_c'') to the cloud. (ID_c, X_c'', Y_c'') is stored in the record of TA. The cloud publishes (Y_c', Y_c'') as its full public key and keeps (X_c', X_c'') as its full private key. Similarly, for a participant $U_n (n \in [1, N])$, it first randomly selects $X_n' \in \mathbb{Z}_p^\star$ as its partial private key and calculates $Y_n' = X_n'P$ as its partial public key. Then, it sends its identity ID_n to the TA. TA computes $Y_n'' = H_1(ID_n)$, $X_n'' = X_{TA}Y_n''$ and returns key pair (X_n'', Y_n'') to the U_n. (ID_n, X_n'', Y_n'') is stored in the record of TA. The participant U_n publishes (Y_n', Y_n'') as its full public key and secretly keeps (X_n', X_n'') as its full private key.
3) *Authentication:* For each participant U_n, before uploading the updated model parameters W_n to the cloud, it generates the signature σ_n and signs W_n as follows.
   - Randomly pick $r, \upsilon \in \mathbb{Z}_p^\star$.
   - Calculate $D_n = H_2(W_n||Y_n'||Y_n''||Y_c')$.
   - Calculate $k_n = H_2(W_n||Y_n'||Y_n''||Y_c', rP)$.
   - Calculate $U_n = X_n'Y_{TA} - X_n''k_n$.
   - $\sigma_n = (k_n, U_n)$.
   - Calculate $Z_n = \upsilon P$, $Z_n' = \upsilon Y_c'$.

- Calculate $W_n^\star = Enc_s(W_n||Y_n'||Y_n''||Y_c', Z_n)$.
- Send the update request message $req_n = (W_n^\star, \sigma_n, k_n, U_n, D_n, Z_n')$ to the cloud.

After receiving the request $req_n = (W_n^\star, \sigma_n, k_n, U_n, D_n, Z_n')$, the cloud authenticate the message as follows.

- Recover $Z_n = X_c'^{-1} Z_n'$.
- Calculate $W_n||Y_n'||Y_n''||Y_c' = Dec_s(W_n^\star, Z_n)$.
- Check $D_n \stackrel{?}{=} H_2(W_n||Y_n'||Y_n''||Y_c')$. If the equation holds, continue to the next step. Otherwise, the authentication fails and stop the procedure.
- Check the validity of σ_n following the Verify step in the proposed PCLS. It rejects the request if σ_n is invalid. Otherwise, continue to the next step.
- The cloud establishes a session key $S_n^1 = H_3(D_n||W_n||Z_n)$ for the next first session. For the subsequent sessions, the session key is calculated as $S_n^{j+1} = H_3(S_n^j||Z_n), \forall j \geq 1$.

It is noted that, with the content of W_n, the cloud can detect the data poisoning attacks and free-rider attacks by integrating with the existing detection schemes [8], [9].

4) *Batch Verification:* The cloud can verify the validity of all received signatures via batch verification. Given a list of requests $\mathcal{R} = \{req_n = (W_n^\star, \sigma_n, k_n, U_n, D_n, Z_n')\}_{n=1}^N$, the batch verification follows

$$e(\sum_{n=1}^{N} Y_n', Y_{TA}) \stackrel{?}{=} e(\sum_{n=1}^{N} U_n, P) e(\sum_{n=1}^{N} k_n Y_n'', Y_{TA}). \tag{4}$$

If Eq. (4) holds, all the signatures in the list are valid. Otherwise, the cloud launches the DBV algorithm to find an appropriate batch size and redo batch verification. As shown in Algorithm 1, given the outcome of failure in batch verification, the cloud picks a verification rate ϵ, where $\epsilon \in (0, 1)$ is a percentage value, and randomly selects $M = \epsilon N$ requests from the list $\mathcal{R}$. The M requests are batch verified following Eq. (4). If Eq. (4) holds, the cloud then uses the M messages to aggregate and update the global ML model. Otherwise, the cloud repeats the above steps at most η iterations. If the verification fails in the all η iterations, the cloud continues to lower the batch size with $M = \epsilon M$ and repeats the lines $3 - 15$ in Algorithm 1 until the batch size M is less than λN, where $\lambda \in (0, 1)$ and λN defines the threshold of batch size.

Algorithm 1 Dynamic Batch Verification (DBV)

Input: $\mathcal{R} = \{req_n = (W_n^\star, \sigma_n, k_n, U_n, D_n, Z_n')\}_{n=1}^N$.
Output: A list of legitimate requests $\mathcal{L}$.
Initialization: λ, η, ϵ, $\mathcal{L} = \emptyset$, $M = N$.

1: $M = \epsilon M$;
2: **while** $M \geq \lambda N$ **do**
3: **for** $i = 1, 2, \ldots, \eta$ **do**
4: Randomly pick M requests from the list $\mathcal{R}$;
5: Verify the M requests following Eq. (4);
6: **if** Eq. (4) holds **then**
7: $\mathcal{L} = \{req_n, \forall n \in M\}$.
8: break.
9: **end if**
10: **end for**
11: **if** $\mathcal{L} \neq \emptyset$ **then**
12: break.
13: **else**
14: $M = \epsilon M$.
15: **end if**
16: **end while**
17: **return** $\mathcal{L}$.

VI. Correctness and Security Analysis

In this section, we first provide the correctness proof of the proposed PCLS and AAPBV. We then provide formal proof of unforgeability, confidentiality, and unlinkability through the three theorems in the following. Based on the three theorems, we further analyze the required security properties in the design goals.

A. Correctness Proof

Theorem 1: The signature in PCLS is correct. The batch verification in AAPBV is correct.

Proof: See Appendix A. ■

B. Security Properties

Theorem 2: The proposed PCLS is unforgeable against chosen message attacks under the CDHP assumption.

Proof: See Appendix B. ■

Theorem 3: The proposed AAPBV is confidential against chosen ciphertext attacks under the DDHP assumption.

Proof: See Appendix C. ■

Theorem 4: The proposed AAPBV achieves unlinkability under the DLP assumption.

Proof: See Appendix D. ■

1) Confidentiality: As proved in Theorem 3, the proposed protocol achieves confidentiality under DDHP assumption. In addition, the cloud can reveal the participant's public key (Y_n', Y_n'') given the request message. By checking $D_n \stackrel{?}{=} H_2(W_n||Y_n'||Y_n''||Y_c)$, the cloud can determine whether the message is altered. Therefore, both confidentiality and integrity are achieved in the proposed AAPBV.

2) Anonymity: During the whole process, the participants' real identities are not involved in authentication and data transmissions. Although a participant's partial public key Y_n'' is generated from its real identity, the cloud and other participants cannot recover the real identity from Y_n'' due to the one-way property of the hash function H_1. Therefore, the proposed protocol achieves anonymity.

3) Mutual Authentication: On the one hand, the cloud can authenticate a participant with its private key X_c' and the participant's public key (Y_n', Y_n''). Particularly, the cloud

recovers Z_n with X'_c and obtains (Y'_n, Y''_n) with Z_n. Then the cloud can determine the validity of the signature following Eq. (3). On the other hand, the participant can authenticate the cloud. Since a session key is established for the next transmission, the participant can verify that the message is from the cloud as long as it can decrypt the message with the session key. Therefore, mutual authentication is achieved.

In addition, the proposed protocol is secure from MITM attack. Assume there is an attacker between the cloud and a participant U_n. When the U_n sends the request message req_n to the cloud, the attacker may eavesdrop and alter req_n. By changing the values of W_n, v, and r, the attacker may generate an altered request which can be verified successful by the cloud. The cloud then calculates a session key based on the modified W_n, encrypts the message with the session key, and sends back to the attacker. Subsequently, the attacker may forward the message to the participant. However, for the participant U_n, it can verify that the message is not from the cloud, since its session key is invalid in decrypting the message. Moreover, only the cloud is able to recover the correct W_n, D_n, and Z_n with its private key X'_c and generate the correct session key. Thus, the attacker cannot impersonate the cloud. Therefore, MITM attack is prevented.

4) Unlinkability: As proved in Theorem 4, the malicious participants in the system cannot link the messages to a specific participant. It is noted that since the cloud can recover a participant's public key, it may link a request to a participant based on its public key. However, due to the anonymity, the cloud is not able to reveal its real identity. Therefore, unlinkability is achieved from the view of malicious participants.

5) Unforgeability: As proved in Theorem 2, the proposed PCLS achieves unforgeability under CDHP assumption. Since PCLS is the design basis of our proposed protocol, the unforgeability of the proposed protocol can be proved in the similar approach. It is noted that, in our proposed AAPBV, even the TA cannot forge a valid key since TA cannot reveal the participant's partial private key X'. Thus, unforgeability is achieved.

6) Forward Security: After successful authentication, a session key is established as $S_n^{j+1} = H_3(S_n^j || Z_n)$. Even the private key of the participant is compromised and the current session key is revealed, the previous session keys cannot be recovered due to the one-way property of the hash function. Therefore, forward security is achieved.

7) Backward Security: A session key is established as $S_n^{j+1} = H_3(S_n^j || Z_n)$ after authentication. As proved in Theorem 3, Z_n is confidential in the communications. Since the session key is generated based on both the current session key and Z_n, even the current session key S_n^j is disclosed, the future session keys are not revealed due to the confidentiality of Z_n. Thus, backward security is achieved.

8) Non-Repudiation: As proved in Theorem 2, a participant uses its signature to sign the message and the signature is unforgeable. Thus, once the message is authenticated, the participant cannot deny that it has sent the message due to the unforgeability of the signature. In addition, when a legal participant is detected to upload false data, the cloud can link the participant with its public key and send the public key to TA. TA can reveal the real identity of the misbehavior participant by searching its records. Therefore, the design goal of non-repudiation is achieved.

TABLE II
OPERATING TIME

Operations	Time (ms)
Exponentiation in $\mathbb{G}_2$	t_e: 1.5
Multiplication in $\mathbb{G}_1$	t_m: 15.175
Pairing	t_p: 18.06

TABLE III
COMPARISON OF THE COMPUTATIONAL TIME WITHOUT BATCH VERIFICATION

Protocols	Sign	Verify	Total
Liu et al. [11]	$5t_m$	t_e + $4t_m$ + t_p	t_e + $9t_m$ +t_p
Zhang et al. [12]	$9t_m$	$6t_m$	$15t_m$
Yang et al. [14]	t_e + t_m	t_e + $5t_p$	$2t_e$ + t_m +$5t_p$
Shen et al. [15]	t_e + t_m + t_p	t_e + t_m +$2t_p$	$2t_e$ + $2t_m$ + $3t_p$
Proposed	$4t_m$	t_e+ $2t_m$ +$3t_p$	t_e + $6t_m$ + $3t_p$

TABLE IV
COMPUTATIONAL TIME OF BATCH VERIFICATION

Protocols	Time
Yang et al. [14]	$5(n+1)t_m$ + $5t_p$
Shen et al. [15]	nt_e + nt_m + $2nt_p$
Proposed	nt_m + $3t_p$

Notes: n represents the batch size.

TABLE V
COMPARISON OF STORAGE COST

Protocols	Storage Cost
Liu et al. [11]	$N(\|ID\| + 2\|K\|)$
Zhang et al. [12]	$N(\|ID\| + 2\|K\|)$
Yang et al. [14]	$N(\|ID\| + 2\|K\| + 2\|C\|)$
Shen et al. [15]	$N(\|ID\| + 2\|K\| + \|C\|)$
Proposed	$N(\|ID\| + 2\|K\|)$

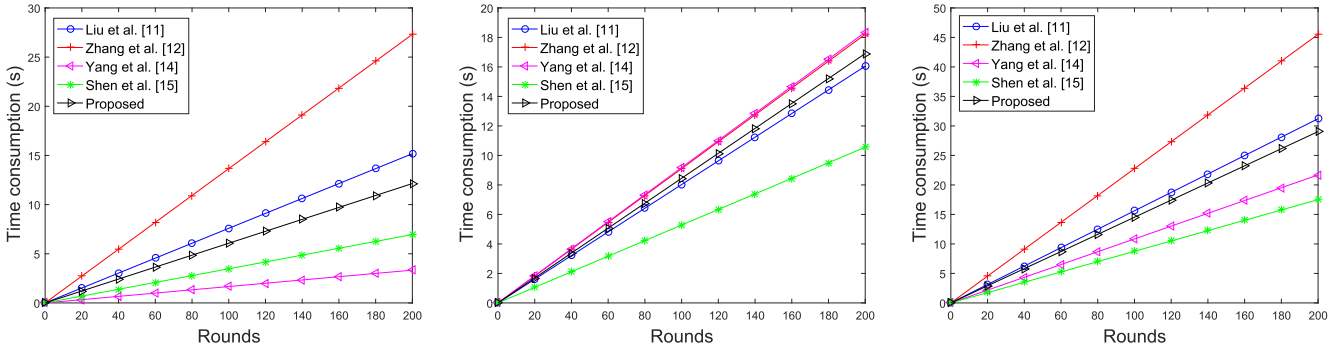
Notes: $|ID|$, $|K|$, and $|C|$ represent the bit sizes of ID, key, and certificate separately. N represents the number of registered participants.

VII. PERFORMANCE EVALUATION

In this section, we compare the computational efficiency and security properties of the proposed AAPBV with [11], [12], [14], [15], [29], and [30]. We implement the proposed protocol on the curve $y^2 = x^3 + x$ with the embedding degree $k = 2$, which achieves the same security level of 1024-bit RSA. The experimental platform is constructed on the cryptographic libraries of PBC [31] and GMP [32] with the Linux system over an Intel(R) Core(TM) i7-4770 3.4GHz processor and 16GB memory. We also evaluate the DBV in our proposed AAPBV by implementing SVM over the dataset BCD [33] in DL. BCD is a breast cancer dataset that involves 570 instances and 32 features. The following numerical results are based on the average values of 100 simulation runs.

A. Computational Efficiency

In the proposed AAPBV, since the operations of exponentiation, multiplication, and pairing dominate the computational



(a) Sign (b) Verify (c) Total

Fig. 3. Comparison of computational time without batch verification.

cost, we evaluate the computational efficiency only based on the three operations. Table II lists the running time of the three operations over the constructed experimental platform, where t_e denotes the time consumption of one exponentiation operation in $\mathbb{G}_2$, t_m denotes the time consumption of one multiplication operation in $\mathbb{G}_1$, and t_p denotes the time consumption of one pairing operation.

Table III and Fig. 3 show the comparison of the computational overhead without batch verification from the theoretical results and simulations respectively. It is noted in Fig. 3 that, our proposed AAPBV outperforms [12] while consuming more time than [11], [14], [15] in different stages. Thus, the advantage of our proposed protocol is not significant comparing with [11], [14], [15] in a single verification. The reason is that to support batch verification and more security features (e.g., forward security and non-repudiation), our proposed protocol consumes more operations of pairing and multiplication.

Table IV and Fig. 4 show the comparison of the computational overhead with batch verification from the theoretical results and simulations respectively. Let the batch size be n. For [15], the computational time for batch verification is $nt_e + nt_m + 2nt_p$, which is the n times of its verification time without batch verification in Table III. Thus, although [15] supports batch verification, the time efficiency is not improved comparing with the case without batch verification. For the proposed AAPBV and [14], the time cost is linearly dependent on the batch size and t_m. Compared with the time consumption listed in Table III, our work and [14] not only support batch verification, but also improve the computational efficiency. As illustrated in Fig. 4, [15] costs the most computational time, while our proposed AAPBV reduces the time consumption significantly. Particularly, in our proposed protocol, the time consumption is reduced by approximately 71% compared with [15] and is reduced by approximately 80% compared with [14]. It is also noted that the batch verification in our proposed protocol reduces the time cost approximately by 82% compared with the case that without batch verification.

Overall, considering Tables III, IV and Figs. 3, 4, our proposed AAPBV has the best time efficiency in batch verification. To support the batch verification, our work employs more operations of pairing and multiplication in single verification,

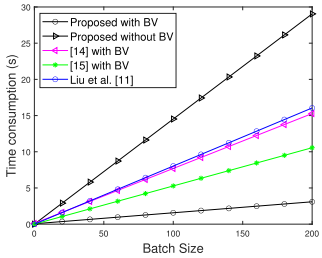


Fig. 4. Computational time with batch verification (BV).

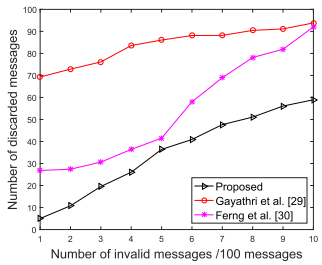


Fig. 5. Performance comparison of batch verification (original batch size is 100).

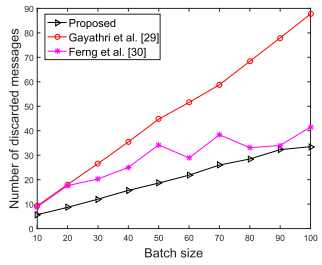


Fig. 6. Performance comparison of batch verification (5/100 messages are invalid).

slightly sacrificing the time efficiency in the case of without batch verification. Since the final authentication is conducted with batch verification, the sacrifice is acceptable.

Figs. 5, 6, 7 illustrate the performance of our DBV by comparing with [29] and [30]. In Fig. 5, we consider the original batch size is 100. In Fig. 6, we consider there are 5 invalid messages among 100 messages. As shown, the

TABLE VI
FEATURES COMPARISON

Features	Liu et al. [11]	Zhang et al. [12]	Yang et al. [14]	Shen et al.[15]	Proposed
Confidentiality	✓	✓	✓	✓	✓
Anonymity	✓	✓	×	✓	✓
Mutual Authentication	✓	✓	×	×	✓
Unlinkability	✓	✓	×	×	✓
Unforgery	✓	×	✓	×	✓
Forward Security	×	✓	×	×	✓
Backward Security	×	×	×	×	✓
Non-repudiation	✓	×	×	×	✓
Batch Verification	×	×	✓	✓	✓

Notes: "✓" represents "support the feature"; "×" represents "not support the feature".

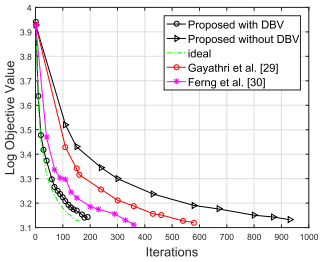


Fig. 7. Convergence iterations.

number of discarded messages is reduced significantly in our proposed protocol. Since less messages are discarded in batch verification, the global model is converged with less iterations as shown in Fig. 7.

Table V shows the comparison of the storage cost at TA. Assume that the key size is $|K|$ bits, the certificate size is $|C|$ bits, the ID size is $|ID|$ bits, and the number of registered participants is N. Then for [14] and [15], the storage cost at TA is $N(|ID|+2|K|+2|C|)$ and $N(|ID|+2|K|+|C|)$ separately. Whereas, [11], [12], and the proposed AAPBV have the same storage cost as $N(|ID|+2|K|)$. Thus, the proposed AAPBV has less storage cost than [14] and [15]. The reduced storage cost is benefited from the certificateless signatures. It is also noted that the proposed AAPBV has better time efficiency than [11] and [12] as illustrated in Figs. 3, 4.

B. Security Properties

Table VI shows the features comparison of the proposed AAPBV with the other work. Compared with our proposed AAPBV, [11] and [12] can satisfy most of the security properties in our design goals. However, neither of them is able to support batch verification. For [14] and [15], although both of them support batch verification, they cannot satisfy all the desired security properties. The table demonstrates that our proposed AAPBV outperforms the others, since it can satisfy all the desired security properties while supporting batch verification.

In addition, benefiting from the achieved security properties, the proposed AAPBV can be integrated with the existing defense schemes [7]–[9] to detect the data poisoning attacks and free-rider attacks. Specifically, after receiving the request req_n and revealing W_n, the cloud can deploy the detection algorithms in [7]–[9] by analyzing the plaintext of W_n. Since the cloud has access to the accurate content of W_n, the detection accuracy should be as good as that in [7]–[9]. Moreover, due to the achieved non-repudiation property, once a participant is detected to upload poisoned ML models, the TA can reveal its real identity based on its public key. Therefore, the integrated scheme is promising in enabling the cloud to detect the data poisoning attacks and free-rider attacks while protecting the participants' privacy.

To summary, the above performance evaluation shows:

- Our proposed protocol achieves efficient batch verification with slightly sacrifice of computational efficiency in single verification. In addition, benefiting from the designed DBV algorithm, the global model converges quickly under our proposed protocol, further improving the computational efficiency in DL.
- Our proposed protocol guarantees various security properties while supporting batch verification. It is a crucial result that our work provides high efficiency and enhanced security with slightly sacrifice of computational costs in single authentication, being more practical for large-scale DL systems.

VIII. CONCLUSION

In this paper, we have proposed an anonymous and efficient authentication protocol as a novel approach to solve the privacy issues in DL. To achieve this, we have designed a certificateless signature scheme PCLS which is proved to be secure in unforgeability. By employing PCLS, we have further designed the protocol AAPBV which supports dynamic batch verification. Security analysis demonstrates that AAPBV satisfies confidentiality, anonymity, mutual authentication, unlinkability, unforgeability, forward security, backward security, and non-repudiation. The performance analysis has shown that AAPBV achieves high computational efficiency and comprehensive security properties.

In the future work, we will extend our work in three directions. 1) We will extend our proposed scheme by integrating with the detailed defense schemes to detect the internal attacks (e.g., data poisoning attack and free-rider attack) in DL. 2) In this work, the security of unlinkability is built under DLP assumption. From the perspective of mathematics, this security property could be further improved. We will construct the unlinkability under other hard problem assumption

(e.g., DDHP, CDHP, etc), such that stronger security will be achieved. 3) Once a malicious participant is detected or a participant quits the system, the TA should remove the participant from the system and revoke the corresponding key issued by TA. Establishing a revocation list at TA side and periodically updating the list with the cloud could be a possible solution. The cloud can filter malicious messages by checking the revocation list before authentication. However, for the large-scale DL, the list could be long. Then checking the revocation list would be time-consuming. Thus, time efficiency should be considered when designing the revocation scheme.

APPENDIX A
PROOF OF THEOREM 1

Proof: The correctness of the signature in PCLS can be proved as follows:

$$
\begin{aligned}
& e(U, P)e(Y'', Y_{TA})^k && \text{(5a)}\\
&= e(U, P)e(Y'', X_{TA}P)^k && \text{(5b)}\\
&= e(U, P)e(X'', P)^k && \text{(5c)}\\
&= e(U + kX'', P) && \text{(5d)}\\
&= e(X'Y_{TA}, P) && \text{(5e)}\\
&= e(Y_{TA}, Y') && \text{(5f)}\\
&= e(Y', Y_{TA}). && \text{(5g)}
\end{aligned}
$$

The correctness of the batch verification in AAPBV is proved as follows:

$$
\begin{aligned}
& e(\sum_{n=1}^{N} U_n, P)e(\sum_{n=1}^{N} k_n Y_n'', Y_{TA}) && \text{(6a)}\\
&= e(\sum_{n=1}^{N} U_n, P)e(\sum_{n=1}^{N} k_n Y_n'', X_{TA}P) && \text{(6b)}\\
&= e(\sum_{n=1}^{N} U_n, P)e(\sum_{n=1}^{N} k_n X_n'', P) && \text{(6c)}\\
&= e(\sum_{n=1}^{N} (U_n + k_n X_n''), P) && \text{(6d)}\\
&= e(\sum_{n=1}^{N} X_n' Y_{TA}, P) && \text{(6e)}\\
&= e(Y_{TA}, \sum_{n=1}^{N} Y_n') && \text{(6f)}\\
&= e(\sum_{n=1}^{N} Y_n', Y_{TA}). && \text{(6g)}
\end{aligned}
$$

■

APPENDIX B
PROOF OF THEOREM 2

Proof: Suppose $\mathcal{A}$ is a probabilistic polynomial time (PPT) adversary who is capable to break the proposed PCLS with non-negligible probability. $\mathcal{C}$ is a challenger who aims to compute abP given a CDHP instance (P, aP, bP). $\mathcal{C}$ can obtain the solution of the CDHP instance by playing the following interactive game with $\mathcal{A}$. Note that the random oracle is a machine with black box that outputs truly random numbers that are independent identically distributed.

1) *Setup Phase*
   $\mathcal{C}$ randomly picks $X \in \mathbb{Z}_p^\star$ as its private key and computes $Y = XP$ as its public key. $\mathcal{C}$ gives $(l, p, P, e, \mathbb{G}_1, \mathbb{G}_2, H_1, H_2, Y)$ and $(ID, Y'' = H_1(ID), X'' = Y''P)$ to $\mathcal{A}$, where H_1, H_2 are random oracles controlled by $\mathcal{C}$.
2) *Query Phase*
   In this phase, $\mathcal{C}$ initializes two empty lists L_1 and L_2 to maintain the responses of hash queries. $\mathcal{A}$ interactively performs a set of queries with $\mathcal{C}$ as follows.
   - H_1 Query: When $\mathcal{A}$ queries the random oracle H_1 with an input $\perp$, $\mathcal{C}$ responses as follows.
     a) If $\perp$ is in the item of H_1, $\mathcal{C}$ searches the item $(\perp, Y'')$ in H_1 and outputs Y''.
     b) Otherwise, $\mathcal{C}$ randomly selects $Y'' \in \mathbb{Z}_p^\star$, adds $(\perp, Y'')$ to H_1, and outputs Y''.
   - Key-Extract Query: $\mathcal{A}$ can query the partial private key given the public key Y''. $\mathcal{C}$ then computes $X'' = XY''$ and outputs X'' to $\mathcal{A}$.
   - H_2 Query: When $\mathcal{A}$ queries the random oracle H_2 with an input $\top$, where $\top$ is a couple of (h, Q) given $h \in \{0,1\}^\star$, $Q \in \mathbb{G}_1$, $\mathcal{C}$ responses as follows.
     a) If $\top$ is in the item of H_2, $\mathcal{C}$ searches the item $(\top, k)$ in H_1 and outputs k.
     b) Otherwise, $\mathcal{C}$ randomly selects $k \in \mathbb{Z}_p^\star$, adds $(\top, k)$ to H_2, and outputs k.
   - Sign Query: When $\mathcal{A}$ queries the signature oracle with message m, $\mathcal{C}$ simulates and responds the query as follows.
     a) $\mathcal{C}$ randomly selects $U, Q \in \mathbb{G}_1$ and $k \in \mathbb{Z}_p^\star$, where k is not equal to any existing output of H_2 oracle.
     b) If $\top = (m, Q)$ has been queried in H_2 oracle, continue to the previous one step. Otherwise, $\mathcal{C}$ adds $(\top, k)$ to H_2.
     c) $\mathcal{C}$ outputs (k, U) as the signature of message m.
3) *Forgery Phase*
   Based on the query phase, a valid signature (k, U) can be generated without knowing the partially private key X'. Thus, in this phase, $\mathcal{A}$ can generate a signature $\sigma = (k, U)$ for message m, where k is generated through the H_2 query and m has never been queried in the sign query. $\mathcal{C}$ tries to obtain the solutions for the CDHP instance as follows.
   a) By repeating the process, $\mathcal{A}$ can produce two valid signatures $\sigma^1 = (k^1, U^1)$, $\sigma^2 = (k^2, U^2)$ and send to $\mathcal{C}$. Then $\mathcal{C}$ obtains
   $$\begin{cases} U^1 + X'' = k^1 X'Y, \\ U^2 + X'' = k^2 X'Y. \end{cases} \quad (7)$$
   b) $\mathcal{C}$ receives $\eta = X'Y = (k^1 - k^2)^{-1}(U^1 - U^2)$ derived from Eq. (7). With (η, X''), $\mathcal{C}$ can generate

a valid signature $\sigma = (k, U)$ for message m just as a real signer using (X', X''), where $k \in \mathbb{Z}_p^\star$ and $U = k\eta - X''$.

c) Then $\mathcal{C}$ can obtain $X'XP = X'Y = \eta$, where $Y = XP, Y' = X'P$. In other words, given a CDHP instance $(P, X'P, XP)$, $\mathcal{C}$ obtains $X'XP = \eta$.

However, the above results obviously contradict the hardness of CDHP, since based on the assumption of CDHP, it is intractable to solve within polynomial time. In other words, the assumption, that $\mathcal{A}$ is a PPT adversary who is capable to break the proposed PCLS with non-negligible probability, is not true. Thus, the theorem is proved.

■

APPENDIX C
PROOF OF THEOREM 3

Proof: Suppose $\mathcal{A}$ is a PPT adversary who is capable to break the proposed AAPBV protocol with non-negligible probability. $\mathcal{C}$ is a challenger who aims to determine weather $abP \equiv Q$ given a DDHP instance (aP, bP, Q). The solution of the DDHP instance can be obtained given the following interactive games between $\mathcal{A}$ and $\mathcal{C}$.

Setup:

The same as in the proof of Theorem 2.

Game 0: This game models the original CCA on the AAPBV protocol.

1) *Query Phase A*
In this phase, $\mathcal{C}$ initializes an empty list L_R to maintain the query responses. $\mathcal{A}$ interactively performs a set of queries with $\mathcal{C}$ as follows.
   - Extract Query: When inputting an index i by $\mathcal{A}$, $\mathcal{C}$ chooses a random value $r_i \in \mathbb{Z}_p^\star$ and generates (r_i, R_i, R_i'), where $R_i = r_iP$, $R_i' = r_iY$. $\mathcal{C}$ records (r_i, R_i, R_i') into the list L_R.
   - R Query: When $\mathcal{A}$ inputs any one term from (r_i, R_i, R_i'), $\mathcal{C}$ returns the corresponding other terms to $\mathcal{A}$.
   - Encryption Query: $\mathcal{A}$ inputs message m and any one term from (r_i, R_i, R_i'), which is from the list L_R. $\mathcal{C}$ returns $m^\star = Enc_s(m, R_i)$ to $\mathcal{A}$.
   - Decryption Query: $\mathcal{A}$ inputs $m^\star$ and any one term from (r_i, R_i, R_i'), which is from the list L_R. $\mathcal{C}$ returns $m = Dec_s(m^\star, R_i)$ to $\mathcal{A}$.
2) *Challenge Phase*
At some point, $\mathcal{A}$ decides to finish query phase A. Then it sends two same-length messages m_0, m_1 and one term from (r_i, R_i, R_i') to $\mathcal{C}$, where the term has never been queried in R Query. $\mathcal{C}$ calculates and returns the challenge ciphertext $Enc_s(m_{\mathbf{b}}, R_i)$ to $\mathcal{A}$. $\mathbf{b}$ is a random bit in $\{0, 1\}$.
3) *Query Phase B*
It is almost the same as that in query phase A, except for the following restriction.
   a) *R Query:* The input term has never been queried to this oracle.
4) *Guess*: $\mathcal{A}$ outputs a guess $\mathbf{b}'$ on $\mathbf{b}$. $\mathcal{A}$ wins the game if $\mathbf{b}' = \mathbf{b}$. The $\mathcal{A}$'s advantage of winning the game is

$$Adv_{\mathcal{A}}^{CCA} = \left| Pr[\boldsymbol{b}' = \boldsymbol{b}] - \frac{1}{2} \right|. \tag{8}$$

If $Adv_{\mathcal{A}}^{CCA}$ is negligible, then the proposed protocol is CCA secure.

Game 1: In this game, we modify Game 0 as follows.

Given (aQ, bQ, V) as the DDHP instance, $\mathcal{C}$ sets $P_1 = aQ = P$, $P_2 = bQ$.

1) *Query Phase A*
In this phase, $\mathcal{C}$ initializes an empty list L_R to maintain the query responses. $\mathcal{A}$ interactively performs a set of queries with $\mathcal{C}$ as follows.
   - Extract Query: When inputting an index i by $\mathcal{A}$, $\mathcal{C}$ chooses a random value $r_i \in \mathbb{Z}_p^\star$ and a value $\theta_i \in \{0, 1\}$ with $Pr[\theta_i = 1] = \delta$. If $\theta_i = 1$, $\mathcal{C}$ sets $R_i = r_iP_1$. If $\theta_i = 0$, $\mathcal{C}$ sets $R_i = r_iP_2$. At last, $\mathcal{C}$ records (r_i, R_i, R_i') into the list L_R.
   - R Query: The same as the Q query in Game 0.
   - Encryption Query: $\mathcal{A}$ inputs message m and any one term from (r_i, R_i, R_i'), which is from the list L_R. If $\theta_i = 1$, $\mathcal{C}$ returns $m^\star = Enc_s(m, R_i)$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ returns "Fail" and exits the game.
   - Decryption Query: $\mathcal{A}$ inputs $m^\star$ and any one term from (r_i, R_i, R_i'), which is from the list L_R. If $\theta_i = 1$, $\mathcal{C}$ returns $m = Dec_s(m^\star, R_i)$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ returns "Fail" and exits the game.
2) *Challenge Phase*: At some point, $\mathcal{A}$ decides to finish the query phase. Then it sends two same-length messages m_o, m_1, and one term from (r_i, R_i, R_i') to $\mathcal{C}$, where the term has never been queried in R Query. If $\theta_i = 1$, $\mathcal{C}$ calculates and returns the challenge ciphertext $Enc_s(m_{\mathbf{b}}, R_i)$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ returns "Fail" and exits the game.
3) *Query Phase B*
It is almost the same as that in query phase A of Game 0, except for the following restriction.
   a) *R Query:* The input term has never been queried to this oracle.
4) *Guess*: $\mathcal{A}$ outputs a guess $\mathbf{b}'$ on $\mathbf{b}$. $\mathcal{A}$ wins the game if $\mathbf{b}' = \mathbf{b}$. Assume the number of queries that $\mathcal{A}$ makes in each oracle of query phase is n_{ex}, n_R, n_{enc}, n_{dec}. Then the probability that $\mathcal{C}$ does not "Fail" is $\delta^{n_{ex}+n_R+n_{enc}+n_{dec}+1}$. The advantage that $\mathcal{A}$ solves the DDHP is

$$Adv_{\mathcal{A}}^{DDHP} = Adv_{\mathcal{A}}^{CCA}\delta^{n_{ex}+n_R+n_{enc}+n_{dec}+1}. \tag{9}$$

Based on the equation (9), if $Adv_{\mathcal{A}}^{CCA}$ is non-negligible, then so does $Adv_{\mathcal{A}}^{DDHP}$. However, this is contradiction because of the hardness of DDHP. In other words, $Adv_{\mathcal{A}}^{CCA}$ is negligible. Therefore, the theorem is proved.

■

APPENDIX D
PROOF OF THEOREM 4

Proof: Suppose $\mathcal{A}$ is a PPT adversary who can distinguish two different participants with non-negligible probability given

two request messages. $\mathcal{C}$ is a challenger who aims to find an integer $n \in \mathbb{Z}_p^\star$ such that $V = nQ$ given a DLP instance (V, Q). $\mathcal{C}$ can obtain the solution of the DLP instance by playing the following interactive game with $\mathcal{A}$.

1) *Setup Phase*
The same as in the proof of Theorem 2.
2) *Query Phase*
In this phase, $\mathcal{C}$ initializes three empty lists L_1, L_2, L_R to maintain the responses of hash queries. $\mathcal{A}$ interactively performs a set of queries with $\mathcal{C}$ as follows.
   - H_1 Query: The same as in the proof of Theorem 2.
   - H_2 Query: The same as in the proof of Theorem 2.
   - Key-Extract Query: The same as in the proof of Theorem 2.
   - Extract Query: The same as in the proof of Theorem 3.
   - R Query: The same as in the proof of Theorem 3.
   - Sign Query: When $\mathcal{A}$ queries the signature oracle with message m, $\mathcal{C}$ simulates and responds the query as follows.
     a) $\mathcal{C}$ randomly selects $U, Q \in \mathbb{G}_1$ and $k, r \in \mathbb{Z}_p^\star$, where k is not equal to any existing output of H_2 oracle, r is not equal to any existing output of R oracle,
     b) If $\top = (m, Q)$ has been queried in H_2 oracle, continue to the previous one step. Otherwise, $\mathcal{C}$ adds $(\top, k)$ to H_2.
     c) $\mathcal{C}$ computes $R' = rY$ and outputs (k, U, R') as the request for message m.
3) *Challenge Phase*
Based on the query phase, a valid request (k, U, R') can be generated without knowing the partially private key X', where k has never been generated through H_2 query. R' has never been generated through extract query. $\mathcal{C}$ tries to obtain the solutions for the DDHP instance as follows.
   a) By repeating the process, $\mathcal{A}$ can produce two valid requests (k_1, U_1, R_1') and (k_2, U_2, R_2') and send to $\mathcal{C}$.
   b) $\mathcal{C}$ obtains
$$\begin{cases} R_1 = X^{-1} R_1', \\ R_2 = X^{-1} R_2'. \end{cases} \tag{10}$$
   Derived from Eq. (10), $\mathcal{C}$ can receive $R_1 - R_2 = X^{-1}(R_1' - R_2')$.
   c) $\mathcal{C}$ can obtain $V = X^{-1}Q$, where $V = R_1 - R_2$, $Q = R_1' - R_2'$. In other words, given a DLP instance (V, Q), $\mathcal{C}$ can find an integer $X^{-1} \in \mathbb{Z}_p^\star$ such that $V = X^{-1}Q$ whenever X^{-1} exists.

However, the above results obviously contradict the hardness of DLP, since based on the assumption of DLP, it is intractable to solve within polynomial time. In other words, the assumption, that $\mathcal{A}$ is a PPT adversary who is capable to break the proposed protocol with non-negligible probability, is not true. Thus, the Theorem is proved. ■

REFERENCES

[1] L. Zhou, D. Wu, X. Wei, and Z. Dong, "Seeing isn't believing: QoE evaluation for privacy-aware users," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 7, pp. 1656–1665, Jul. 2019.

[2] S. Gyawali, Y. Qian, and R. Hu, "Deep reinforcement learning based dynamic reputation policy in 5G based vehicular communication networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 6136–6146, Jun. 2021.

[3] Y. Jiang, K. Zhang, Y. Qian, and L. Zhou, "Reinforcement-learning-based query optimization in differentially private IoT data publishing," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11163–11176, Jan. 2021.

[4] K. A. Bonawitz *et al.*, "Towards federated learning at scale: System design," in *Proc. Mach. Learn. Syst.*, Palo Alto, CA, USA, 2019, pp. 374–388.

[5] Z. He, T. Zhang, and R. B. Lee, "Attacking and protecting data privacy in edge–cloud collaborative inference systems," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9706–9716, Jun. 2021.

[6] O. A. Wahab, A. Mourad, H. Otrok, and T. Taleb, "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1342–1397, Feb. 2021.

[7] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh, and S. Yu, "PoisonGAN: Generative poisoning attacks against federated learning in edge computing systems," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3310–3322, Mar. 2021.

[8] J. Chen, X. Zhang, R. Zhang, C. Wang, and L. Liu, "De-Pois: An attack-agnostic defense against data poisoning attacks," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3412–3425, 2021.

[9] Y. Fraboni, R. Vidal, and M. Lorenzi, "Free-rider attacks on model aggregation in federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 1846–1854.

[10] H. H. Kilinc and T. Yanik, "A survey of SIP authentication and key agreement schemes," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1005–1023, Oct. 2013.

[11] J. Liu, Z. Zhang, X. Chen, and K. S. Kwak, "Certificateless remote anonymous authentication schemes for wirelessbody area networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 332–342, Feb. 2014.

[12] A. Zhang, L. Wang, X. Ye, and X. Lin, "Light-weight and robust security-aware D2D-assist data transmission protocol for mobile-health systems," *IEEE Trans. Inf. Forensics, Secur.*, vol. 12, no. 3, pp. 662–675, Mar. 2017.

[13] F. Qu, Z. Wu, F.-Y. Wang, and W. Cho, "A security and privacy review of VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 2985–2996, Dec. 2015.

[14] A. Yang, X. Tan, J. Baek, and D. S. Wong, "A new ADS-B authentication framework based on efficient hierarchical identity-based signature with batch verification," *IEEE Trans. Services Comput.*, vol. 10, no. 2, pp. 165–175, Mar./Apr. 2017.

[15] J. Shen, D. Liu, X. Chen, J. Li, N. Kumar, and P. Vijayakumar, "Secure real-time traffic data aggregation with batch verification for vehicular cloud in VANETs," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 807–817, Jan. 2020.

[16] H. Jiang, J. Pei, D. Yu, J. Yu, B. Gong, and X. Cheng, "Applications of differential privacy in social network analysis: A survey," *IEEE Trans. Knowl. Data Eng.*, early access, Apr. 13, 2021, doi: 10.1109/TKDE.2021.3073062.

[17] T. Zhu, G. Li, W. Zhou, and S. Y. Philip, "Differentially private data publishing and analysis: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 8, pp. 1619–1638, Aug. 2017.

[18] C. Dwork, "Differential privacy," in *Proc. Int. Colloq. Automata, Lang. Program. (ICALP)*, 2006, pp. 1–12.

[19] T. Zhang and Q. Zhu, "Dynamic differential privacy for ADMM-based distributed classification learning," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 172–187, Jan. 2017.

[20] W. Y. B. Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, Apr. 2020.

[21] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "Local differential privacy for deep learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5827–5842, Jul. 2020.

[22] H. Zheng, H. Hu, and Z. Han, "Preserving user privacy for machine learning: Local differential privacy or federated machine learning?" *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 5–14, Jul. 2020.

[23] X. Wang, J. He, P. Cheng, and J. Chen, "Privacy preserving collaborative computing: Heterogeneous privacy guarantee and efficient incentive mechanism," *IEEE Trans. Signal Process.*, vol. 67, no. 1, pp. 221–233, Jan. 2019.
[24] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics, Security*, vol. 13, no. 5, pp. 1333–1345, May 2018.
[25] P. Mohassel and Y. Zhang, "SecureML: A system for scalable privacy-preserving machine learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 19–38.
[26] Q. Jia, L. Guo, Y. Fang, and G. Wang, "Efficient privacy-preserving machine learning in hierarchical distributed system," *IEEE Trans. Netw. Sci. Eng.*, vol. 6, no. 4, pp. 599–612, Oct. 2019.
[27] C. Zhou, A. Fu, S. Yu, W. Yang, H. Wang, and Y. Zhang, "Privacy-preserving federated learning in fog computing," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 10782–10793, Apr. 2020.
[28] J. Huang, D. Fang, Y. Qian, and R. Q. Hu, "Recent advances and challenges in security and privacy for V2X communications," *IEEE Open J. Veh. Technol.*, vol. 1, pp. 244–266, 2020.
[29] N. B. Gayathri, G. Thumbur, P. V. Reddy, and M. Z. U. Rahman, "Efficient pairing-free certificateless authentication scheme with batch verification for vehicular ad-hoc networks," *IEEE Access*, vol. 6, pp. 31808–31819, 2018.
[30] H.-W. Ferng, J.-Y. Chen, M. Lotfolahi, Y.-T. Tseng, and S.-Y. Zhang, "Messages classification and dynamic batch verification scheme for VANETs," *IEEE Trans. Mobile Comput.*, vol. 20, no. 3, pp. 1156–1172, Mar. 2021.
[31] *The Pairing-Based Cryptography Library*. Accessed: Jul. 30, 2021. [Online]. Available: https://crypto.stanford.edu/pbc/
[32] *The Gnu Multiple Precision Arithmetic Library*. Accessed: Jul. 30, 2021. [Online]. Available: https://gmplib.org/
[33] *The Breast Cancer Dataset*. Accessed: Jul. 30, 2021. [Online]. Available: https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/

Yili Jiang (Student Member, IEEE) received the B.S. degree in electrical and information engineering from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2014. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Nebraska–Lincoln. Her research interests include cyber security, information privacy, intelligent Internet of Things, and machine learning.

Kuan Zhang (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Canada, in 2016. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of Nebraska–Lincoln, Lincoln, NE, USA. He has published over 100 papers in journals and conferences. His research interests include cyber security, big data, the Internet of Things, and cloud/edge computing. He was a recipient of the Best Paper Award from IEEE WCNC 2013, SECURECOMM 2016, and ICC 2020.

Yi Qian (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Clemson University, Clemson, SC, USA, in 1996. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Nebraska–Lincoln (UNL). His research interests include communications and systems, and information and communication network security. He serves on the Editorial Boards of several international journals and magazines, including as the Editor-in-Chief for IEEE WIRELESS COMMUNICATIONS. He was previously the Chair of the IEEE Technical Committee for Communications and Information Security. He was the Technical Program Chair of the IEEE International Conference on Communications in 2018. He was a Distinguished Lecturer of the IEEE Vehicular Technology Society and the IEEE Communications Society.

Liang Zhou (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from École Normale Supérieure (ENS), Cachan, France, and Shanghai Jiao Tong University, Shanghai, China, in 2009. He is currently a Professor at the Nanjing University of Posts and Telecommunications, China. His research interests include multimedia communications and computing.