An Efficient Conditional Privacy-Preserving Authentication Scheme for Vehicular Ad Hoc Networks Using Online/Offline Certificateless Aggregate Signature

Kang Li[1,2], Man Ho Au[2,⋆], Wang Hei Ho[3], and Yi Lei Wang[2]

[1] Research Institute for Sustainable Urban Development, The Hong Kong Polytechnic University, Hong Kong
kang.li@connect.polyu.hk

[2] Department of Computing, The Hong Kong Polytechnic University, Hong Kong
man-ho-allen.au@polyu.edu.hk, yilei.wang@polyu.edu.hk

[3] Department of Electronic and Information Engineering,The Hong Kong Polytechnic University, Hong Kong
ivanwh.ho@polyu.edu.hk

Abstract. Vehicular ad hoc networks (VANETs) are fundamental components of building a safe and intelligent transportation system. However, due to its wireless nature, VANETs have serious security and privacy issues that need to be addressed. The conditional privacy-preserving authentication protocol is one important tool to satisfy the security and privacy requirements. Many such schemes employ the certificateless signature, which not only avoids the key management issue of the PKI-based scheme but also solves the key escrow problem of the ID-based signature scheme. However, many schemes have the drawback that the computational expensive bilinear pairing operation or map-to-point hash function are used. In order to enhance the efficiency, certificateless signature schemes for VANETs are usually constructed to support signature aggregation or online/offline computation. In this paper, we propose an efficient conditional privacy-preserving authentication scheme using an online/offline certificateless aggregate signature, which does not require bilinear pairing or map-to-point hash function, to address the security and privacy issues of VANETs. Our proposed scheme is proven to be secure with a rigorous security proof, and it satisfies all the security and privacy requirements with a better performance compared with other related schemes.

1 Introduction

Thanks to the rapid advancement of wireless technologies, the vehicular ad-hoc network (VANET) is introduced to build a safe and intelligent transportation system in metropolitan cities. In VANET, drivers can get a better awareness of their driving environment and can take early action to respond to an emergent situation to avoid any possible damage or to follow a better route by circumventing traffic bottleneck. However, the transmitted message, which may include sensitive data concerning the drivers' privacy, in DSRC wireless protocol could be easily monitored, altered and forged. For example, a malicious vehicle may broadcast a fake message to cause a traffic accident.

⋆ Corresponding author.

For message security, the receiver should verify the legitimacy and integrity of the received message before taking further action. In terms of the privacy issue, anonymity must be provided to prevent the adversaries from extracting private information, such as the real identity, from the transmitted messages. However, privacy protection should be conditional, as traceability should also be guaranteed, which indicates that the TA should be able to reveal the real identity of a malicious vehicle when it is necessary.

Many privacy-preserving authentication schemes based on traditional public key infrastructure (PKI) [12, 19] have been proposed to address the security and privacy issues. However, in PKI-based authentication scheme, a certificate is required for every public key of the vehicle and the RSU, which means that a certificate authority needs ma to manage all the certificates and vehicles may have to preload a large number of public/private key pairs together with the corresponding certificates in the local storage. This causes huge storage burden and also makes it difficult for the authority. Due to this drawback, PKI-based scheme is not practical and still infeasible for use in VANETs. In order to remove the burden of certificates, papers such as [3, 11], proposed ID-based authentication scheme to enhance the computation and communication efficiency. However, these mechanisms are considered suitable only for private networks, because of the key escrow problem [10]. To solve the key escrow problem of ID-based signature scheme, the concept of certificaletess signature was firstly introduced by Al-Riyami and Paterson [1].Since then, many authentication schemes using certificateless signatures have been proposed to tackle the security and privacy problems in VANET [5,14,16,27].

Since the OBU only has limited computation capacity and the communication window of VANET is very short, participants in VANETs need to handle a large flow of messages. Hence, aggregate signature is proposed to improve message authentication efficiency in vanet. Signature aggregation means that given n signatures on n distinct messages from n distinct users, it is possible to aggregate all these signatures into a single short signature [4]. This is very useful in the scenario, where RSUs aid the communications in VANET by collecting and aggregating a large set of individual signatures of each vehicle into one signature and broadcasting this aggregated signature to the vehicles, which greatly enhances the efficiency of verification and reduces the communication overhead. Apart from the aggregated signature, an online/offline signature is another approach to further decrease the computation cost. In the offline phase, some heavy computations are executed and the intermediate results are stored in resource-constrained devices. Then in the online phase, on receiving a message, the device can very efficiently compute a signature using the intermediate result from the offline phase.

In this paper, we propose an efficient pairing-free online/offline aggregated certificateless signature scheme with conditional privacy-preserving for VANETs. Our scheme satisfies all the security and privacy requirements for VANETs with a rigorous security proof. In order to further enhance authentication efficiency, our scheme supports online/offline signing, signature aggregation, and batch verification. Moreover, we analyse its computation efficiency, specifically the signing, verifying and aggregated verifying cost and make comparisons with some other similar schemes to demonstrate that the efficiency of our scheme is better than most of other related schemes.

1.1 Related Works

The introduction of the first certificateless signature (CL-PKS) by Al-Riyami and Paterson [1] has inspired a large body of research work on improving the CL-PKS scheme. Yum and Lee [25] described a general method to construct a CL-PKS scheme from any ID-based signature scheme. Later, Li et al. [15] proposed the first CL-PKS scheme using bilinear pairings. Au et al. [2] presented a new security model for CL-PKS schemes, in which a malicious KGC attack is considered. He et al. [7] developed the first CL-PKS without using bilinear pairings. However, in [22], the scheme in [7] is found to be insecure against a strong type II attack. More recently, Yeh et al.[24] proposed a CL-PKS scheme for IoT deployment. However, Jia et al. [13] pointed out that it has security flaws, as any malicious KGC can impersonate the KGC and it cannot resist a public key replacement attack.

The first online/offline signature scheme was introduced by Even, Goldreich and Micali [6]. But, the method is impractical since the size of the signature increases by a quadratic factor [17]. Liu et al. [17] proposed an efficient identity based online/offline signature scheme, but it has the key escrow problem. Recently, Cui et al.[5] proposed an efficient certificateless aggregated signature scheme without pairing for VANETs. However, Kamil et al. [14] found a security flaw in [5].

2 Preliminaries and Background

2.1 Elliptic Curve Cryptosystem and Assumptions

Let F_p be a finite field, which is determined by a λ-bit prime number p. Let a set of elliptic curve points E over F_p be defined by the curve form: $y^2 = x^3 + ax + b$, where $p > 3$, $a, b \in F_p$, and $(4a^3 + 27b^2)$ mod p$\neq$ 0, and the point at infinity be O. All the points on E including O form an additive group G with order q and generator P. The point addition '+' of element in cyclic group G is defined as follows: Let $P, Q \in G$, l be the line containing P, Q (tangent line to E if $P = Q$), and R is the third point of the intersection of l and E. Let l' be the line connecting R and O. Then P '+' Q is defined as the third point such that l' intersects with E at R and O, which is -R. Scalar multiplication over E/F_p can be defined as follows:

$$mP{=}P{+}P{+}P{+}...{+}P \text{ (m times), where } m \in Z_q^*$$

The following complexity assumptions are used in security proof of the proposed scheme. We will use the Discrete Logarithm (DL) assumption and the Computational Diffie-Hellman (CDH) assumption over the additive cyclic group G, which can be defined as follows.

Definition 1 (The DL Assumption).

Discrete Logarithm (DL) Assumption: Given a random point $Q \in G$ on E, it is hard to compute an integer $x \in Z_q^*$ in polynomial time such that $Q = xP$ with non-negligible probability.

Definition 2 (The CDH Assumption).

Computational Diffie-Hellman (CDH) Assumption: Given two random point $Q, R \in G$ on E, where $Q = xP$, $R = yP$, $x, y \in Z_q^*$, it is hard to compute xyP in polynomial time with non-negligible probability.

2.2 System Model

Typically, a two-layer vehicular ad hoc network model is suitable for VANETs. Fig. 1 shows the typical architecture of VANETs. The lower layer composed of vehicles

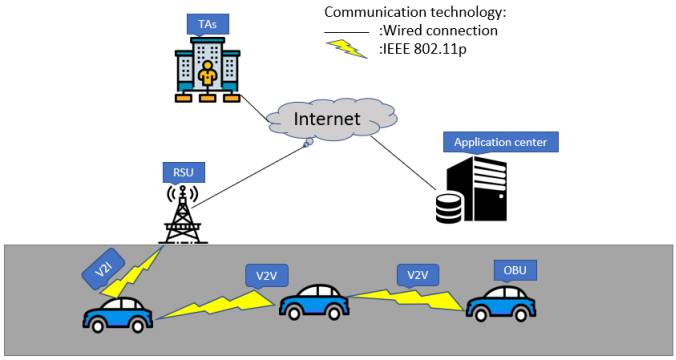

Fig. 1 A typical architecture of VANETs

and roadside units (RSUs) located at the critical points along the road. Each vehicle is equipped with an onboard unit (OBU), which enables vehicles to communicate with other vehicles or RSUs. The communication of Vehicle-to-Everything(V2X), mainly the Vehicle-to-Vehicle(V2V) and Vehicle-to-Infrastructure (V2I), is realized by the dedicated short-range communications (DSRC) protocol, which is identified as IEEE 802.11p. The upper layer of VANET consists of an application server(such as traffic control and analysis center), and key generation center (KGC) and trace authority (TRA). The TRA is responsible for RSU and vehicle registration by generating pseudo identities for them and can reveal the real identity of a vehicle from its signed message. The KGC is in charge of generating public and private keys for RSU and vehicles. Besides, we assume that the KGC and TRA are always trusted and cannot be comprised, which is usually assumed in VANET scheme as in [18, 26]. The KGC and TRA have sufficient computation power and storage capacity. KGC and TRA are two separate authorities, which can communicate with each other securely using wired networks and secure protocols, such as Transport Layer Security(TLS) protocol. We also assume that each vehicle is equipped with a tamper-proof device, which can prevent the adversary from extracting data from the device. The OBU only has limited computation power, and RSU has greater computation power than OBU. The OBU and RSU are not trusted, and the message sent by them should be authenticated.

3 The Proposed Authentication Scheme

In this section, we present our proposed authentication scheme in detail. First, we define some notations that will be used in the scheme as listed in Table1.

Table 1: Notations and Descriptions

Notation	Description
V_i	The i-th vehicle
psk_i	A partial private key of vehicle V_i
x_{ID_i}	A secret key of vehicle V_i
vpk_{ID_i}	A public key of vehicle V_i
(P_{pub}, α)	The public/private key pair of KGC
(T_{pub}, β)	The public/private key pair of TRA
RID_i	The real identity of a vehicle V_i
PID_i	The pseudo identity of a vehicle V_i
H_1, H_2, H_3	Secure hash functions
T_i	A valid period of the pseudo identity
t_i	A current timestamp
m_i	A traffic-related message
$\oplus$	The exclusive **OR** operation
$\|\|$	The message concatenation operation

3.1 System Parameter Setup

In this phase, the TRA and KGC will generate the system parameters, such as a finite field, an elliptic curve, public keys, etc.

- Given a security parameter τ, the TAs will generate two large primes p and q, and will choose a non-singular elliptic curve E, which is defined by the equation $y^2 = x^3 + ax + b$, where $p > 3$, $a, b \in F_p$, and $(4a^3 + 27b^2)$ mod p$\neq 0$.
- The TAs will choose a generator P of the additive group G with the order of q. And it will also choose three secure hash functions which are H_1: $G \times \{0,1\}^* \times \{0,1\}^* \rightarrow Z_q^*$, H_2: $\{0,1\}^* \times G \rightarrow Z_q^*$, H_3: $\{0,1\}^* \times \{0,1\}^* \times G \times G \times \{0,1\}^* \rightarrow Z_q^*$.
- The TRA will randomly choose number $\beta \in Z_q^*$ as its master private key for traceability, and compute $T_{pub} = \beta \cdot P$ as its public key.
- The KGC will randomly choose number $\alpha \in Z_q^*$ as its master private key for partial private key extraction, and compute $P_{pub} = \alpha \cdot P$ as its public key.
- Then, the public parameters are $params$=$\{P, p, q, E, G, H_1, H_2, H_3, P_{pub}, T_{pub}\}$. Finally,each vehicle pre-loads the public parameters into its temper-proof device and RSU stores $params$ into its local storage.

3.2 Pseudo-Identity-Generation and Partial-Private-Key-Extraction

In this phase, vehicles register with the TRA and KGC to obtain its pseudo identity and partial private key.

- The vehicle choose a random value $k_i \in Z_q^*$, and calucate $PID_{i,1}$=k_iP.Then the vehicle sends its real identity RID_i and $PID_{i,1}$ to the TRA in a secure way.
- Once the TRA receives $(RID_i, PID_{i,1})$ from the vehicle, it first check whether RID_i is valid or not. If RID_i exist in its local database, then TRA computes $PID_{i,2}$ =

$RID_i \oplus H_1((\beta \cdot PID_{i,1}) \| T_i \| T_{pub})$ and send the $PID_{i,2}$ to the vehicle. Then, the pseudo identity of the vehicle is PID_i=$(PID_{i,1}, PID_{i,2}, T_i)$ where T_i is the valid period of the pseudo identity.

- A vehicle will use its pseudo identity PID_i to communicate with other participants in the VANET. Since only TRA know its master private key β, it has the ability to reveal the real identity of a vehicle by cumputing $RID_i = PID_{i,2} \oplus H_1((\beta \cdot PID_{i,1}) \| T_i \| T_{pub})$ in some situation. Then, the TRA will also send the pseudo identity PID_i to KGC in a secure way.
- After the KGC receives the pseudo identity, it choose a random number $d_i \in Z_q^*$ and compute Q_{ID_i}=$d_i P$. Then it calculates the partial private key as $psk_{ID_i} = d_i + H_2(PID_i \| Q_{ID_i}) \cdot \alpha \pmod q$.
- Then the KGC transmits (Q_{ID_i}, psk_{ID_i}) to the vehicle via a secure channel. Finally the vehicle obtains its pseudo identity PID_i and partial private key psk_{ID_i}. And the vehicle can check the validity of the partial private key using the public parameters by verifying whether the equation $psk_{ID_i} \cdot P$=$Q_{ID_i} + H_2(PID_i \| Q_{ID_i}) \cdot P_{pub}$ holds or not. If it holds, then the vehicle will store the pseudo identity (PID_i) and partial private key(psk_{ID_i}) in its temper-proof device for further use. Note that the value Q_{ID_i} should be public.

3.3 Vehicle-Key-Generation

In this phase, the vehicle choose a random number $x_{ID_i} \in Z_q^*$ as its secret key and compute $vpk_{ID_i} = x_{ID_i} \cdot P$ as its public key.

3.4 Offline-Sign

In order to maintain the message authentication and integrity, the traffic-related message should be signed before transmitted. Since the computation power of the OBU is limited, we propose to use online-offline signature technique, which allows the vehicles to offline compute some part of the signature when OBU is idle or the traffic density is not high, to enhance the efficiency of generating signatures. The offline signature is generated as follows:

- V_i randomly selects a number $r_i \in Z_q^*$
- V_i computes $R_i = r_i \cdot P$
- V_i stores the offline $\phi_i = (r_i, R_i)$ locally

Generating the offline signature does not require the message, thus a large set of these offline signature pairs could be pre-generated and stored locally for future use.

3.5 Online-Sign

Firstly, it randomly picks a pseudo identity PID_i from its storage and selects the latest timestamp t_i, which is used to prevent the replay message attacks. On input a traffic-related message m_i, it signs the message as the followings steps.

- V_i obtains a fresh offline signature tuple $\phi_i = (r_i, R_i)$ from its storage.
- V_i computes the full private key $sk_i = x_{ID_i} + psk_{ID_i}$

- V_i computes $h_{3i} = H_3(m_i||PID_i||vpk_{ID_i}||R_i||t_i)$.
- V_i computes $s_i = h_{3i} \cdot r_i + sk_i \ (modq)$
- The output signature is $\sigma_i = (R_i, s_i)$. Finally, the vehicle V_i broadcasts $\{m_i, PID_i, \sigma_i, t_i, vpk_{ID_i}, Q_{ID_i}\}$ to nearby RSUs and vehicles for verification.

3.6 Individual-Verify

In this phase, RSUs or vehicles verify the validity of an individual received message. Once it receives the message $\{m_i, PID_i, \sigma_i, t_i, vpk_{ID_i}, Q_{ID_i}\}$, it checks the validity of the signature as follows.

- Firstly, the verifier will check the freshness of the timestamp t_i. If it is not fresh, then the verifier reject the message and stop the verifying process.
- Then, calculate $h_{3i} = H_3(m_i||PID_i||vpk_{ID_i}||R_i||t_i)$ and $h_{2i} = H_2(PID_i||Q_{ID_i})$
- Then,check whether the equation $s_i \cdot P = h_{3i} \cdot R_i + vpk_{ID_i} + Q_{ID_i} + h_{2i} \cdot P_{pub}$ holds or not. If this equation holds, then the verifier acepts this message, otherwise reject.

Proof of Correctness: Since $h_{3i} = H_3(m_i||PID_i||vpk_{ID_i}||R_i||t_i)$, $h_{2i} = H_2(PID_i||Q_{ID_i})$, $sk_i = x_{ID_i} + psk_{ID_i}$, $r_i \cdot P = R_i$, $x_{ID_i} \cdot P = vpk_{ID_i}$, and $psk_{ID_i} \cdot P = Q_{ID_i} + h_{2i} \cdot P_{pub}$, if the signature is generated correctly, then the following equation will hold

$$\begin{aligned} s_i \cdot P &= h_{3i} \cdot r_i \cdot P + x_{ID_i} \cdot P + psk_{ID_i} \cdot P \\ &= h_{3i} \cdot R_i + vpk_{ID_i} + Q_{ID_i} + h_{2i} \cdot P_{pub} \end{aligned}$$

3.7 Aggregate

In some scenarios where the density of transmitted messages is very high, RSUs need to aid the communication by aggregating a collection of certificateless signatures into one. Signature aggregation is the process that on receiving a set of messages $\{m_i, PID_i, \sigma_i, t_i, vpk_{ID_i}, Q_{ID_i}\}$ from n vehicles $\{V_i, V_2, ..., V_n\}$, where $i = 1, 2, 3, ...n$, the RSU aggregate the signature by calculating $S = \sum_{i=1}^{n} s_i$. Then RSUs output $\sigma = (R_1, R_2, R_3...R_n, S)$ as the aggregated signature.

3.8 Aggregate-Verify

This algorithm is assumed to be performed by RSUs or the application centers, such as a traffic control center. Once receiving the aggregated signature $\sigma = (R_1, R_2, R_3...R_n, S)$ from a set of vehicles $\{V_1, V_2, V_3, ..., V_n\}$, with the corresponding parameters $\{m_i, PID_i, t_i, vpk_{ID_i}, Q_{ID_i}\}$, where $i = 1, 2, 3, ...n$,the RSUs or application centers check the validity of the aggregated signature by performing the following steps.

- Firstly, the verifier will check the freshness of the timestamp t_i, for $i = 1, 2, 3, ...n$. If it is not fresh, then the verifier reject the message and stop the verifying process.
- Caculate $h_{3i} = H_3(m_i||PID_i||vpk_{ID_i}||R_i||t_i)$ and $h_{2i} = H_2(PID_i||Q_{ID_i})$, for $i = 1, 2, 3, ...n$
- Check whether the following equation holds or not: $S \cdot P = \sum_{i=1}^{n}(h_{3i} \cdot R_i) + \sum_{i=1}^{n} Q_{ID_i} + \sum_{i=1}^{n} vpk_{ID_i} + (\sum_{i=1}^{n} h_{2i}) \cdot P_{pub}$. If this equation holds, the verifier will accept the aggregated signature.

Proof of Correctness:

Since we have $h_{3i} = H_3(m_i||PID_i||vpk_{ID_i}||R_i||t_i)$, $h_{2i} = H_2(PID_i||Q_{ID_i})$, $sk_i = x_{ID_i} + psk_{ID_i}$, $r_i \cdot P = R_i$, $x_{ID_i} \cdot P = vpk_{ID_i}$, and $psk_{ID_i} \cdot P = Q_{ID_i} + h_{2i} \cdot P_{pub}$, then we can check the correctness as follows:

$S \cdot P = \sum_{i=1}^{n} s_i \cdot P$
$= \sum_{i=1}^{n} (h_{3i} \cdot r_i \cdot P + x_{ID_i} \cdot P + psk_{ID_i} \cdot P)$
$= \sum_{i=1}^{n} (h_{3i} \cdot R_i) + \sum_{i=1}^{n} Q_{ID_i} + \sum_{i=1}^{n} vpk_{ID_i} + (\sum_{i=1}^{n} h_{2i}) \cdot P_{pub}$

3.9 Batch Verification

Sometimes, a participant in VANETs needs to verify multiple signatures in a single instance instead of aggregating them. In this scenario, we need to use the batch verification technique, which allows multiple signatures to be verified at a time. To ensure the non-repudiation of signatures using batch verification, we use the small exponent test technology [11]. On receiving multiple messages $\{m_i, PID_i, \sigma_i, t_i, vpk_{ID_i}, Q_{ID_i}\}$ where $i = 1, 2, 3, ...n$, the verifier checks the signature validity using public parameters. The verification process is presented as follows.

- Firstly, the verifier will check the freshness of the timestamp t_i, for $i = 1, 2, 3, ...n$. If it is not fresh, then the verifier reject the message and stop the verifying process.
- The verifier randomly choose a vector v=$\{v_1, v_2, v_3, ..., v_n\}$,, where v_i is a small random integer in $[1, 2^t]$ and t is a small integer that incurs very little computation head.
- The verifier checks whether the following equation hols, if it holds, it accepts the messages, otherwise rejects the messsages.
  $(\sum_{i=1}^{n} s_i \cdot v_i) \cdot P = \sum_{i=1}^{n} (h_{3i} \cdot R_i \cdot v_i) + \sum_{i=1}^{n} (vpk_{ID_i} \cdot v_i) + \sum_{i=1}^{n} (Q_{ID_i} \cdot v_i) + (\sum_{i=1}^{n} h_{2i} \cdot v_i) \cdot P_{pub}$

Proof of Correctness: The process is similar to that in the aggregated verify. We have $h_{3i} = H_3(m_i||PID_i||vpk_{ID_i}||R_i||t_i)$, $h_{2i} = H_2(PID_i||Q_{ID_i})$, $sk_i = x_{ID_i} + psk_{ID_i}$, $r_i \cdot P = R_i$, $x_{ID_i} \cdot P = vpk_{ID_i}$, and $psk_{ID_i} \cdot P = Q_{ID_i} + h_2 \cdot P_{pub}$. We obtain that:

$(\sum_{i=1}^{n} s_i \cdot v_i) \cdot P$
$= \sum_{i=1}^{n} ((h_{3i} \cdot r_i + x_{ID_i} + psk_{ID_i}) \cdot v_i) \cdot P$
$= \sum_{i=1}^{n} (h_{3i} \cdot v_i \cdot r_i \cdot P) + \sum_{i=1}^{n} (v_i \cdot x_{ID_i} \cdot P) + \sum_{i=1}^{n} (v_i \cdot psk_{ID_i} \cdot P)$
$= \sum_{i=1}^{n} (h_{3i} \cdot R_i \cdot v_i) + \sum_{i=1}^{n} (psk_{ID_i} \cdot v_i) + \sum_{i=1}^{n} ((Q_{ID_i} + h_{2i} \cdot P_{pub}) \cdot v_i)$
$= \sum_{i=1}^{n} (h_{3i} \cdot R_i \cdot v_i) + \sum_{i=1}^{n} (vpk_{ID_i} \cdot v_i) + \sum_{i=1}^{n} (Q_{ID_i} \cdot v_i) + (\sum_{i=1}^{n} h_{2i} \cdot v_i) \cdot P_{pub}$

4 Security Proof

In this section, we give a formal security proof on the proposed certificateless signature scheme. We use a similar approach in [7] to prove the security of the proposed signature scheme. The detailed security proof is shown in the appendix.

5 Discussion

In this section, we first present the security and privacy analysis with respect to the identity privacy-preserving, message authentication, and integrity, traceability, unlinkability and resistance to various attacks. Then we will analyze the performance of the proposed online/offline certificateless signature scheme and compare with some other similar schemes.

5.1 Security Analysis

1. **Identity Privacy Preserving**: Each participant in VANET needs to register with the TRA to obtain a pseudo identity, which is generated by the TRA using its master private key β. The only way for an adversary to reveal the real identity is to compute $RID_i = PID_{i,2} \oplus H_1((\beta \cdot PID_{i,1})||T_i||T_{pub})$, which means that the adversary has to know the master private key β to calculate $\beta \cdot PID_{i,1}$. However, it is infeasible for the adversary to obtain β from $T_{pub} = \beta \cdot P$, as this contradicts the DL assumption. Therefore, our scheme meets the requirement of identity privacy preserving.
2. **Message Authentication and Integrity**: Each transmitted message is signed by a legitimate user before broadcasting in VANET. According to Theorem 1, and Theorem 2, there is no polynomial-time adversary can forge a valid signature based on the DL assumption. Hence the verifier can check the validity and integrity of the signature, which guarantees that the message comes from a legitimate user and it is not modified during transmission, by verifying the equation $s_i \cdot P = h_{3i} \cdot R_i + vpk_{ID_i} + Q_{ID_i} + h_{2i} \cdot P_{pub}$. Hence, the proposed scheme ensures the message authentication and integrity.
3. **Traceability**: The pseudo identity is generated using the master private key of the TRA. From the pseudo identity $PID_i=(PID_{i,1}, PID_{i,2}, T_i)$, where $PID_{i,1}=k_iP$, $PID_{i,2} = RID_i \oplus H_1((\beta \cdot PID_{i,1})||T_i||T_{pub})$, the TRA can extract the real identity by computing $RID_i = PID_{i,2} \oplus H_1((\beta \cdot PID_{i,1})||T_i||T_{pub})$. Hence, the traceability is also provided by our scheme.
4. **Unlinkability**: During the pseudo identity generation phase, the OBU choose a random value $k_i \in Z_q^*$ to calculate $PID_{i,1}=k_iP$ and $PID_{i,2} = RID_i \oplus H_1((\beta \cdot PID_{i,1})||T_i||T_{pub})$ which compose the pseudo identity. As for the signature generation, a random value $r_i \in Z_q^*$ is also selected by the vehicle and used to compute the signature. Due to the randomness of k_i and r_i, it is infeasible for the adversary to link two anonymous identities or signatures generated by the same vehicle. Hence, the requirement of unlinkability is also guaranteed by our scheme.
5. **Resistance to Various Attacks**: In this part, we show that our scheme can resist various attacks, including reply attack, modification attack, impersonation attack and stolen verifier table attack.
   - **Reply Attack**: The timestamp t_i inside the message $\{m_i, PID_i, \sigma_i, t_i, vpk_{ID_i}, Q_{ID_i}\}$ is used to resist the reply attack. Before verifying the validity of the signature, the verifier will check the freshness of the timestamp t_i. If it is not a fresh timestamp, the message will be rejected. Hence, the reply attack is avoided in our scheme by using the timestamp.
   - **Message Modification Attact**: Since each message is signed by the sender, any modification of the message will lead to the result that equation $s_i \cdot P = h_{3i} \cdot R_i + vpk_{ID_i} + Q_{ID_i} + h_{2i} \cdot P_{pub}$ does not hold when the verifier checks the validity of the signature. Then the modified message will be disregarded. Hence, our scheme can resist modification attack.
   - **Impersonation Attack:** In order to launch a successful impersonation attack, the adversary should be able to output a message $\{m_i, PID_i, \sigma_i, t_i, vpk_{ID_i}, Q_{ID_i}\}$ that can pass the verification of the receiver. This means that the adversary should be able to forge a valid signature. However, this is infeasible according

to the Theorem 1 and Theorem 2. Hence the impersonation attack is impossible for our scheme.

– **Stolen Verifier Table Attack:** In our scheme, OBU and RSU does not maintain a verifier table for message authentication. Therefore, stolen verifier table attack is also impossible for our scheme.

5.2 Performance Evaluation

We adopt a similar approach in [9] to analyze the performance. Below we define the benchmark and security level for comparisons.

For bilinear pairing-based authentication schemes, we use a bilinear pairing $\bar{e} : G_1 \times G_1 \rightarrow G_2$ with the security level of 80-bits, where G_1 is an additive group generated by a point $\bar{P}$ with the order of $\bar{q}$ on the super singular elliptic curve $\bar{E} : y^2 = x^3 + x \bmod \bar{p}$ with the embedding group degree 2, $\bar{p}$ is a 512-bit prime number, $\bar{q}$ is a 160-bit Solinas prime number and the equation $\bar{p} + 1 = 12\bar{q}r$ holds. For ECC-based identity-based authentication scheme, we achieve the security level of 80-bits by using an additive group G generated by a point P with the order q on a non-singular elliptic curve E, which is defined by the equation $y^2 = x^3 + ax + b$, where $p > 3$, $a, b \in F_p$, p, q are 160-bit prime number, and $(4a^3 + 27b^2)$ mod p$\neq 0$.

5.3 Computation Cost Analysis

We first define some notations about the execution time of the cryptographic operations. The execution time is evaluated using the famous MIRACL cryptographic library. We use the cryptographic operation time directly from [9] to evaluate the performance. Note that some very light operations, such as addition operation in Z_q^* and multiplication operation in Z_q^* are ignored, as the execution time is relatively small.

•T_{bp}: The operation time of a bilinear pairing operation $\bar{e}(P, Q)$, where $\bar{P}, \bar{Q} \in G_1$, 4.2110 milliseconds;

•T_{bp-m}: The operation time of a scalar multiplication $x \cdot \bar{P}$ related to a bilinear pairing, where $\bar{P} \in G_1, x \in Z_{\bar{q}}^*$, 1.7090 milliseconds;

•T_{bp-a}: The operation time of a point addition $\bar{P} + \bar{Q}$ related to a bilinear pairing, where $\bar{P}, \bar{Q} \in G_1$, 0.0071 milliseconds;

•T_{ecc-m}: The operation time of a scalar multiplication $x \cdot P$ related to the ECC, where $P \in G$ and $x \in Z_q^*$, 0.4420 milliseconds;

•T_{ecc-a}: The operation time of a point addition $P + Q$ related to the ECC, where $P, Q \in G$, 0.0018 milliseconds;

•T_H: The execution time of a map-to-point hash function operation, 4.406 milliseconds;

•T_h: The execution time of an ordinary one-way hash function operation, 0.0001 milliseconds.

We make comparisons with the recent authentication schemes in VANET [5, 10, 14, 16, 20, 27]. The comparisons of computation cost of signing, verifying one message and aggregated verify are given in Table 2 and Table 3. From Table 2 and Table 3, it is obvious to see that schemes[10, 16, 20, 27] with pairing operation and map-to-point hash functions are much more computationally expensive than schemes based on ECC

Table 2: Computation cost comparisons of the proposed scheme with others

Schemes	Sign(ms)	Individual Verify(ms)	Total(ms)
[20]	$4T_{bp-m} + 2T_{bp-a} + T_h \approx 6.8503$	$3T_{bp} + 3T_{bp-m} + T_{bp-a} + 2T_h \approx 17.7673$	24.6176
[10]	$2T_{bp-m} + T_{bp-a} + T_h \approx 3.4252$	$3T_{bp} + T_{bp-m} + T_{bp-a} + T_H + T_h \approx 18.7552$	22.1804
[16]	$3T_{bp-m} \approx 5.127$	$3T_{bp} + 2T_H + 2T_{bp-m} \approx 24.863$	29.99
[27]	$3T_{bp-m} \approx 5.127$	$3T_{bp} + T_H + 2T_{bp-m} \approx 20.457$	25.584
[5]	$T_{ecc-m} + T_h + T_{ecc-a} \approx 0.4439$	$3T_{ecc-m} + 2T_{ecc-a} + 2T_h \approx 1.3298$	1.7737
[14]	$3T_{ecc-m} + 3T_h + 2T_{ecc-a} \approx 1.3299$	$2T_{ecc-m} + T_{ecc-a} + T_h \approx 0.8859$	2.2158
Our scheme	$T_{ecc-m} + T_h \approx 0.4421$	$3T_{ecc-m} + 3T_{ecc-a} + 2T_h \approx 1.3316$	1.7737

Table 3: Computation cost comparisons of the proposed scheme with others

Schemes	Aggregated Verify(ms)
[20]	$3T_{bp} + 3nT_{bp-m} + nT_{bp-a} + 2nT_h$
[10]	$3T_{bp} + nT_{bp-m} + nT_{bp-a} + nT_H + nT_h$
[16]	$3T_{bp} + (n+1)T_H + 2nT_{bp-m}$
[27]	$3T_{bp} + nT_H + 2nT_{bp-m}$
[5]	$(n+2)T_{ecc-m} + 2nT_{ecc-a} + 2nT_h$
[14]	$2T_{ecc-m} + nT_{ecc-a} + nT_h$
Our scheme	$(n+2)T_{ecc-m} + 3nT_{ecc-a} + 2nT_h$

cryptographic primitives and simple one-way hash functions. Then, comparing to similar schemes [5,14], which also does not require pairing and map-to-point hash function, our scheme also has some advantages. Even through [5] almost has the same computation efficiency as our scheme, it is shown to be insecure under the existing security model in [14]. Kamil et al. [14] proposed an improved scheme after its cryptanalysis of Cui's scheme [5]. Although, the individual verifying phase of our scheme is more expensive than that in [14], the signing cost of our scheme is much lower than that in [14]. And note that, the total cost of signing and verifying a single message is also small than that in [14]. More importantly, our scheme supports online/offline sign, which means that some cryptographic operations can be pre-computed and used directly when signing a message. Hence in our scheme, the signing cost could be lower and only be T_h, as the operation of the relatively expensive scalar multiplication corresponding to T_{ecc-m} can be pre-computed and does not incur computation overhead.

In Fig.2, we further investigate the aggregated verification time with respect to the number of signatures. Fig.1 indicates that the aggregated verification time with regards to number of signatures of the schemes, which require bilinear pairings and map-to-point hash functions, increases much faster than that of the schemes without pairings or map-to-point hash functions. The aggregated verification time with regards to the number of signatures of our scheme grows a little faster than that of [14]. However, we argue that typically a RSU is assumed to have much more computation power than the OBU. Hence, in many scenarios, the need to enhance the signing efficiency is more

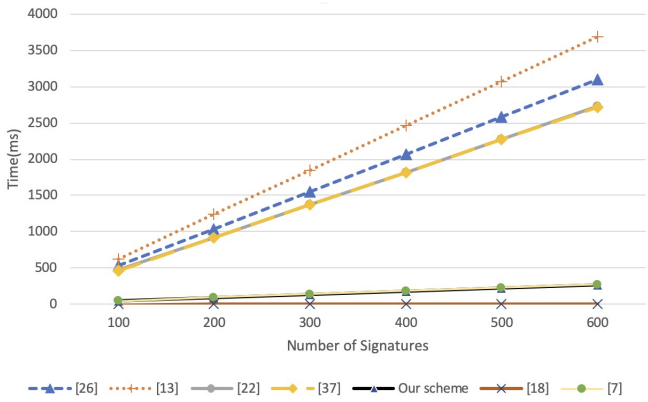

Fig. 2 Aggregated verification time vs. Number of signatures

significant than the need to improve the aggregated verfication efficiency, which means that the advantage of an efficient sign phase outweight the advantage of an efficient aggregated verification phase. Therefore, our scheme has a slight edge comparing to the scheme [14] in the sense that the signing efficiency is higher than that in [14].

6 Conclusions

In this paper, we propose an efficient conditional privacy-preserving authentication scheme using online/offline certificateless aggregate signature to address the security and privacy issues of VANETs. Our proposed scheme is proven to be secure with a rigorous security proof, and it satisfies all the security and privacy requirements of VANET. The online/offline signature allows some computational expensive operations to be pre-computed offline, thus reducing the computation overhead when signing a message online. Moreover, the proposed scheme does not require the computational expensive bilinear pairing operation and map-to-point hash function, and it supports signature aggregation and batch verification, which are very useful for VANETs scenario. As a result of using these techniques, the proposed scheme has a better computation efficiency compared with many other related schemes.

References

[1] S. S. Al-Riyami and K. G. Paterson. Certificateless public key cryptography. In *International conference on the theory and application of cryptology and information security*, pages 452–473. Springer, 2003.

[2] M. H. Au, Y. Mu, J. Chen, D. S. Wong, J. K. Liu, and G. Yang. Malicious kgc attacks in certificateless cryptography. In *Proceedings of the 2nd ACM symposium on Information, computer and communications security*, pages 302–311. ACM, 2007.

[3] M. Bayat, M. Barmshoory, M. Rahimi, and M. R. Aref. A secure authentication scheme for vanets with batch verification. *Wireless networks*, 21(5):1733–1743, 2015.

[4] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 416–432. Springer, 2003.

[5] J. Cui, J. Zhang, H. Zhong, R. Shi, and Y. Xu. An efficient certificateless aggregate signature without pairings for vehicular ad hoc networks. *Information Sciences*, 451:1–15, 2018.

[6] S. Even, O. Goldreich, and S. Micali. On-line/off-line digital signatures. In *Conference on the Theory and Application of Cryptology*, pages 263–275. Springer, 1989.

[7] D. He, J. Chen, and R. Zhang. An efficient and provably-secure certificateless signature scheme without bilinear pairings. *International Journal of Communication Systems*, 25(11):1432–1442, 2012.

[8] D. He, M. Tian, and J. Chen. Insecurity of an efficient certificateless aggregate signature with constant pairing computations. *Information sciences*, 268:458–462, 2014.

[9] D. He, S. Zeadally, B. Xu, and X. Huang. An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *IEEE Transactions on Information Forensics and Security*, 10(12):2681–2691, 2015.

[10] S.-J. Horng, S.-F. Tzeng, P.-H. Huang, X. Wang, T. Li, and M. K. Khan. An efficient certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks. *Information Sciences*, 317:48–66, 2015.

[11] S.-J. Horng, S.-F. Tzeng, Y. Pan, P. Fan, X. Wang, T. Li, and M. K. Khan. b-specs+: Batch verification for secure pseudonymous authentication in vanet. *IEEE Transactions on Information Forensics and Security*, 8(11):1860–1875, 2013.

[12] J.-P. Hubaux, S. Capkun, and J. Luo. The security and privacy of smart vehicles. *IEEE Security & Privacy*, (3):49–55, 2004.

[13] X. Jia, D. He, Q. Liu, and K.-K. R. Choo. An efficient provably-secure certificateless signature scheme for internet-of-things deployment. *Ad Hoc Networks*, 71:78–87, 2018.

[14] I. A. Kamil and S. O. Ogundoyin. An improved certificateless aggregate signature scheme without bilinear pairings for vehicular ad hoc networks. *Journal of information security and applications*, 44:184–200, 2019.

[15] X.-x. Li, K.-f. Chen, and L. Sun. Certificateless signature and proxy signature schemes from bilinear pairings. *Lithuanian Mathematical Journal*, 45(1):76–83, 2005.

[16] D. LIU, R.-h. SHI, S. ZHANG, and H. ZHONG. Efficient anonymous roaming authentication scheme using certificateless aggregate signature in wireless network. *Journal on Communications*, 37(7):182–192, 2016.

[17] J. K. Liu, J. Baek, J. Zhou, Y. Yang, and J. W. Wong. Efficient online/offline identity-based signature for wireless sensor network. *International Journal of Information Security*, 9(4):287–296, 2010.

[18] N.-W. Lo and J.-L. Tsai. An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks without pairings. *IEEE Transactions on Intelligent Transportation Systems*, 17(5):1319–1328, 2015.

[19] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen. Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications. In *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pages 1229–1237. IEEE, 2008.

[20] A. K. Malhi and S. Batra. An efficient certificateless aggregate signature scheme for vehicular ad-hoc networks. *Discrete Mathematics and Theoretical Computer Science*, 17(1):317–338, 2015.

[21] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of cryptology*, 13(3):361–396, 2000.

[22] J.-L. Tsai, N.-W. Lo, and T.-C. Wu. Weaknesses and improvements of an efficient certificateless signature scheme without using bilinear pairings. *International Journal of Communication Systems*, 27(7):1083–1090, 2014.

[23] H. Xiong, Z. Guan, Z. Chen, and F. Li. An efficient certificateless aggregate signature with constant pairing computations. *Information Sciences*, 219:225–235, 2013.
[24] K.-H. Yeh, C. Su, K.-K. R. Choo, and W. Chiu. A novel certificateless signature scheme for smart objects in the internet-of-things. *Sensors*, 17(5):1001, 2017.
[25] D. H. Yum and P. J. Lee. Generic construction of certificateless signature. In *Australasian Conference on Information Security and Privacy*, pages 200–211. Springer, 2004.
[26] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen. An efficient identity-based batch verification scheme for vehicular sensor networks. In *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pages 246–250. IEEE, 2008.
[27] H. Zhong, S. Han, J. Cui, J. Zhang, and Y. Xu. Privacy-preserving authentication scheme with full aggregation in vanet. *Information Sciences*, 476:211–221, 2019.

A Security Proof

Typically, for a certificateless signature scheme, we define two types of security, namely Type-I security and Type-II security, which cooresponds to two types of adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$.

- **Type-I Adversary**: $\mathcal{A}_1$ can launch a public key replacement attack by replacing the public key of any vehicle with a value of its choice. $\mathcal{A}_1$ does not know the master secret key or the partial private key.
- **Type-II Adversary**: $\mathcal{A}_2$ acts as a malicious-but-passive KGC, which knows the master key and the partial private key, but cannot replace any user's public key.

Theorem 1. *The proposed scheme is $(\varepsilon, t, q_c, q_s, q_h)$- secure against the adversary $\mathcal{A}_1$ in the random oracle model, assuming that DL assumption hold in G, where q_c, q_h, q_s are the numbers of **Create**,**Hash** and **Sign** queries that the adversary is allowed to make.*

Proof. Assume there is a probabilistic polynomial-time forger $\mathcal{A}_1$, we construct an algorithm $\mathcal{F}$ that make use of $\mathcal{A}_1$ to solve the discrete logarithm problem(DLP). Suppose $\mathcal{F}$ is given the DLP instance (P, Q) to compute $x \in Z_q^*$ such that $Q = xP$. $\mathcal{F}$ chooses a random identity ID^* as the challenged ID and answers the oracle queries from $\mathcal{A}_1$ as follows:

- **Setup(ID) query:** $\mathcal{F}$ sets $P_{pub} = Q$ and sends the parameters $\{P, p, q, E, G, H_2, H_3, P_{pub}\}$ to $\mathcal{A}_1$.
- **Create(ID) query:** $\mathcal{F}$ maintains a hash list L_c of tuple $(ID, Q_{ID}, vpk_{ID}, psk_{ID}, x_{ID}, h_2)$. When $\mathcal{A}_1$ makes a query on ID, if ID is in L_c, $\mathcal{F}$ responds with $(ID, Q_{ID}, vpk_{ID}, psk_{ID}, x_{ID}, h_2)$. Otherwise, $\mathcal{F}$ will simulate the oracle as follows. It randomly selects three value $a, b, c \in Z_q^*$, and sets $Q_{ID} = a \cdot P_{pub} + b \cdot P$, $vpk_{ID} = c \cdot P$, $psk_{ID} = b, x_{ID} = c, h_2 = H_2(ID||Q_{ID}) \leftarrow -a(modq)$. Then it responds with $(ID, Q_{ID}, vpk_{ID}, psk_{ID}, x_{ID}, h_2)$, and inserts (ID, Q_{ID}, h_2) to L_{H_2}. Note that the equation $psk_{ID} \cdot P = Q_{ID} + h_2 \cdot P_{pub}$ holds, which means that the partial secret key is valid.
- H_2 **query:**When adversary makes a H_2 query with (ID, Q_{ID}), if ID is already in the hash list L_{H_2}, $\mathcal{F}$ just returns the corresponding h_2. Otherwise, $\mathcal{F}$ runs Create(ID) to get h_2, and send h_2 to $\mathcal{A}_1$.

– **Partial-Private-Key-Extract(ID) query:** If $ID = ID^*$, $\mathcal{F}$ stops the simulation. Otherwise, $\mathcal{F}$ checks the hash list L_c, if ID in the list, then $\mathcal{F}$ response with psk_{ID}. If ID is not in L_c, $\mathcal{F}$ queries Create(ID) to get the psk_{ID}, and sends it to $\mathcal{A}_1$.
– **Public-Key(ID) query:** On receiving the query on ID, if ID is already in L_c, $\mathcal{F}$ response with pk_{ID}=(Q_{ID}, vpk_{ID}). Otherwise, $\mathcal{F}$ queries Create(ID) to get the (Q_{ID}, vpk_{ID}), and sends it to $\mathcal{A}_1$.
– **Public-Key-Replacement(ID, pk'_{ID}) query:** $\mathcal{F}$ maintains a hash list L_R of tuple ($ID, d_i, Q_{ID}, x_{ID}, vpk_{ID}$). When $\mathcal{A}_1$ queries with (ID, pk'_{ID}), where $Q'_{ID} = d'_i \cdot P$,$vpk'_{ID} = x'_{ID} \cdot P$ and pk'_{ID} =(Q'_{ID}, vpk'_{ID}), $\mathcal{F}$ sets $Q_{ID} = Q'_{ID}$, $vpk_{ID} = vpk'_{ID}$, $psk_{ID} = \perp$, and $x_{ID} = x'_{ID}$. Then $\mathcal{F}$ updates the list L_R to be ($ID, d'_i, Q'_{ID}, vpk'_{ID}, x_{ID}$)
– H_3 **query:** $\mathcal{F}$ maintains a hash list L_{H_3} of tuple ($m, ID, R, vpk_{ID}, t, h_3$). If the queries ID is in this list, $\mathcal{F}$ just responds with h_3. Otherwise it chooses a random h_3, sets $h_3 = H_3(m||ID||vpk_{ID}||R||t)$, add it into L_{H_3} and responds with h_3.
– **Sign(ID, m) query:** When $\mathcal{A}_1$ makes a sign query on (ID, m), if ID is in L_R, $\mathcal{F}$ generates random numbers $a, b, c \in Z_q^*$, sets $s = a, R = P, h_3 = H_3(m||ID||vpk_{ID}||R||t) \leftarrow (a - b - c)mod(q)$, inserts ($m, ID, R, vpk_{ID}, t, h_3$) into L_{H_3}. The output signature is (R, s). If ID is not in L_R, $\mathcal{F}$ acts like the description of the scheme.

Finally, $\mathcal{A}_1$ outputs a forged signature σ=(R, $s_{\{1\}}$) on (ID, m), which satisfies the verification process of the verifier. If $ID \neq ID^*$,$\mathcal{F}$ fails and aborts. From the forking lemma in [21], $\mathcal{F}$ rewinds $\mathcal{A}_1$ to the point where it queries H_3, and use a different value. $\mathcal{A}_1$ will output another valid signatures (R, $s_{\{2\}}$) with the same R.Then we have:

$$s_{\{i\}} \cdot P = h_{3_{\{i\}}} \cdot R + vpk_{ID} + Q_{ID} + h_2 \cdot P_{pub}, \text{ where } i = 1, 2$$

From these two linear equations, we can derive the value r by $\frac{s_2 - s_1}{h_{3_{\{2\}}} - h_{3_{\{1\}}}}$. Another rewind on H_2 will allow computation on x.

Probability Analysis: The simulation of Create(ID) oracle fails when the random oracle assignment $H_2(ID||Q_{ID})$ causes inconsistency, which happens with the probability at most q_h/q. The probability of successful simulation of q_c times is at least (1-$(q_h/q))^{q_c} \geqq$1-$(q_h q_c/q)$. Also, the simulation is successful q_h times with the probability at least (1-$(q_h/q))^{q_h} \geqq$1-(q_h^2/q). And $ID = ID^*$ with the probability $1/q_c$. Therefore, the overall successful simulation probability is (1-$q_h q_c/q$)(1-(q_h^2/q))$(1/q_c)\varepsilon$.

The time complexity of the algorithm $\mathcal{F}$ is dominated by the exponentiations performed in the Create and Sign queries, which is equal to $t + O(q_c + qs)S$, where S is the time of a scalar multiplication operation.

Theorem 2. *The proposed scheme is ($\varepsilon, t, q_c, q_s, q_h$)- secure against the adversary $\mathcal{A}_2$ in the random oracle model, assuming that DL assumption hold in G, where q_c, q_h, q_s are the numbers of **Create**,**Hash** and **Sign** queries that the adversary is allowed to make.*

Proof. Assume there is a probabilistic polynomial-time forger $\mathcal{A}_2$, we construct an algorithm $\mathcal{F}$ that make use of $\mathcal{A}_2$ to solve the discrete logarithm problem(DLP). Suppose $\mathcal{F}$ is given the DLP instance (P, Q) to compute $y \in Z_q^*$ such that $Q = yP$. $\mathcal{F}$ chooses a random identity ID^* as the challenged ID and answers the oracle queries from $\mathcal{A}_2$ as follows:

– **Setup(ID) query:** $\mathcal{F}$ sets $P_{pub} = x \cdot P, x \in Z_q^*$ and sends the parameters {P, p, q, E, G, H_2, H_3, P_{pub}} to $\mathcal{A}_2$.

- **Create(ID) query:** $\mathcal{F}$ maintains a hash list L_c of tuple ($ID, Q_{ID}, vpk_{ID}, psk_{ID}, x_{ID}$, h_2). When $\mathcal{A}_1$ makes a query on ID, if ID is in L_c, $\mathcal{F}$ responds with (ID, Q_{ID}, $vpk_{ID}, psk_{ID}, x_{ID}, h_2$). If $ID = ID^*$, $\mathcal{F}$ choose $a, b \in Z_q^*$ randomly, sets $Q_{ID} = aP$, $vpk_{ID} = Q, h_2 = H_2(ID||Q_{ID}) \leftarrow b, psk_{ID} = a+x\cdot h_2, x_{ID} = \perp$. If $ID \neq ID^*, \mathcal{F}$ select three random number a, b, c, and sets $Q_{ID} = aP, vpk_{ID} = bP, h_2 = H_2(ID||Q_{ID}) \leftarrow c, psk_{ID} = a + x \cdot h_2, x_{ID} = b$. Finally, $\mathcal{F}$ response the query with ID, Q_{ID}, vpk_{ID}, psk_{ID}, x_{ID}, h_2 and add ID, Q_{ID}, h_2 into the hash list L_{H_2}
- H_2 **query:**When adversary makes a H_2 query with (ID, Q_{ID}), if ID is already in the hash list L_{H_2}, $\mathcal{F}$ just returns the corresponding h_2. Otherwise, $\mathcal{F}$ runs Create(ID) to get h_2, and send h_2 to $\mathcal{A}_1$.
- **Partial-Private-Key-Extract(ID) query:** On receiving the query on ID, $\mathcal{F}$ checks the hash list L_c, if ID in the list, then $\mathcal{F}$ response with psk_{ID}. If ID is not in L_c, $\mathcal{F}$ queries Create(ID) to get the psk_{ID}, and sends it to $\mathcal{A}_1$.
- **Public-Key(ID) query:** On receiving the query on ID, if ID is already in L_c, $\mathcal{F}$ response with pk_{ID}=(Q_{ID}, vpk_{ID}). Otherwise, $\mathcal{F}$ queries Create(ID) to get the (Q_{ID}, vpk_{ID}), and sends it to $\mathcal{A}_1$.
- **Secrety-Key-Extract(ID) query:** If $ID = ID^*$,$\mathcal{F}$ aborts the simulation. Otherwise, if ID is already in L_c, $\mathcal{F}$ response with x_{ID}.If ID is not alreary in L_c, $\mathcal{F}$ runs Create(ID) to get $ID, Q_{ID}, vpk_{ID}, psk_{ID}, x_{ID}, h_2$, and sends $x_{ID}$to the adversary.
- H_3 **query:** $\mathcal{F}$ maintains a hash list L_{H_3} of tuple ($m, ID, R, vpk_{ID}, t, h_3$). If the quries ID is in this list, $\mathcal{F}$ just responds with h_3. Otherwise it chooses a random h_3, sets $h_3 = H_3(m||ID||vpk_{ID}||R||t)$, add it into L_{H_3} and responds with h_3.
- **Sign(ID, m) query:** If $ID \neq ID^*, \mathcal{F}$ acts like the description of the scheme.Otherwise, $\mathcal{F}$ generates random numbers $a, b, f \in Z_q^*$, sets $s = a, h_3 = H_3(m||ID||vpk_{ID}||R||t) \leftarrow f, R = h_3^{-1}\cdot (bP_{pub} - Q)$, and response eith the signature as (R, s). This signature is valid as the euqation $s \cdot P = h_3 \cdot R + Q_{ID} + vpk_{ID} + h_2 \cdot P_{pub}$ holds.

Finally, $\mathcal{A}_2$ outputs a forged signature σ=(R, $s_{\{1\}}$) on (ID, m), which satisfies the verification process of the verifier. From the forking lemma in [21], $\mathcal{F}$ rewinds $\mathcal{A}_2$ to the point where it queries H_3, and use a different value. $\mathcal{A}_2$ will output another valid signature (R, $s_{\{2\}}$) with the same R.Then we have:

$$s_{\{i\}} \cdot P = h_{3_{\{i\}}} \cdot R + vpk_{ID} + Q_{ID} + h_2 \cdot P_{pub}, \text{ where } i = 1, 2$$
$$s_{\{i\}} = h_{3_{\{i\}}} \cdot r + y + d_i + h_2 x, i = 1, 2$$

Only y, r are unknown. Hence, from these two linear equations, we can derive the two unknown value r, y, and output y as the solution of the DL problem.

Probability Analysis: The simulation of Create(ID) oracle fails when the random oracle assignment $H_2(ID||Q_{ID})$ causes inconsistency, which happens with the probability at most q_h/q. The probability of successful simulation of q_c times is at least (1-$(q_h/q))^{q_c} \geqq$1-$(q_h q_c/q)$. Also, the simulation is successful q_h times with the probability at least (1-$(q_h/q))^{q_h} \geqq$1-(q_h^2/q). And $ID = ID^*$ with the probability $1/q_c$. Therefore, the overall successful simulation probabiloity is (1-$q_h q_c/q$)(1-(q_h^2/q))$(1/q_c)\varepsilon$.

The time complexity of the algorithm $\mathcal{F}$ is dominated by the exponentiations performed in the Create and Sign queries, which is equal to $t + O(q_c + qs)S$, where S is the time of a scalar multiplication operation.