Check for updates

IET Information Security

Research Article

IET Journals
The Institution of Engineering and Technology

Secure hierarchical authentication protocol in VANET

ISSN 1751-8709
Received on 16th May 2019
Revised 25th August 2019
Accepted on 26th September 2019
E-First on 12th November 2019
doi: 10.1049/iet-ifs.2019.0249
www.ietdl.org

Xuelian Li[1,2], Yue Han[1] ✉, Juntao Gao[3], Jie Niu[1]

[1]*School of Mathematics and Statistics, Xidian University, Xi'an, Shaanxi, People's Republic of China*
[2]*Guangxi Key Laboratory of Cryptography and Information Security, Guilin, Guangxi, People's Republic of China*
[3]*School of Telecommunications Engineering, Xidian University, Xi'an, Shaanxi, People's Republic of China*
✉ *E-mail: Han_Yue0526@163.com*

Abstract: The 5G technology will promote the development of vehicle *ad hoc* networks (VANET). However, almost all the existing authentication protocols rely on a completely trusted authority (TA), which undoubtedly raises a heavy burden on the TA. On the other hand, these protocols can rarely resist some special attacks, such as registration authority leaks registration information attack and ephemeral secret leakage attack. To address these problems, based on the self-certified public keys and Schnorr signatures, the authors propose a hierarchical revocable authentication protocol in VANET. The proposed protocol is provably secure in the random oracle model under the Diffie-Hellman assumption. Performance analysis illustrates that the protocol greatly saves computation resources and satisfies the well-known security requirements. Therefore, the proposed protocol is suitable for the VANET environment.

Nomenclature

$\hat{e}$	admissible bilinear map from $G_1 \times G_1$ to G_2
P	generator of group G_1
s	system private key of central authority TA_1
P_{pub}	system public key defined by $P_{\mathrm{pub}} = s \cdot P$
U	participant $U \in \{C, S\}$, where C and S denote vehicle and RSUs
ID_U	real identity of participant U
PID_U	anonymous identity of participant U
d_U	private key of participant U
d_{tU}	time key of participant U
H_1	map-to-point hash function
H_i	general one-way hash function
CA_1	single top-layer central authority
RA_2	multiple second-layer registration authorities
$\mathrm{RA}_{\mathrm{ID}_2}$	identity of second-layer registration authority
$P_{\mathrm{RA}_{\mathrm{ID}_2}}$	public key of second-layer registration authority

1 Introduction

With the continuous exploration of 5G technology, the vehicle *ad hoc* network (VANET) has been further developed. 5G, short for the 5th generation mobile networks, is the latest development in cellular technology to increase the speed of wireless networks. The international organisations are deeply involved in the 5G standardisation work now. The 3rd Generation Partnership Project defined three 5G application scenarios, including eMBB (enhanced Mobile Broadband), URLLC (Ultra Reliable Low Latency Communications) and mMTC (massive Machine Type Communications). In short, 5G will bring users higher bandwidth rates, lower and more reliable delays and meet requirements of 'Internet of everything'. Therefore, 5G can solve the communication latency problem in VANET and achieve instant message services.

VANET [1, 2], a vital component of intelligent transportation systems [3], can share information to ease traffic congestion and improve road efficiency. It provides vehicle-to-vehicle communication, vehicle-to-infrastructure communication and so on. Fig. 1 depicts a basic structure of VANET system, which contains a trusted authority (TA), on board units (OBUs) and road side units (RSUs). The TA is not only responsible for vehicle and RSU registration, but also for subsequent revocation. RSUs are fixed roadside infrastructures and combine the functions of sensing, analysis and communication. During the Mobile World Congress Shanghai 2018 conference [4], the world's first RSU commercial solution was launched to accelerate the commercialisation of Internet of vehicles technologies. OBUs are equipped on vehicles to do limited computation, storage and other processing. At the same time, information sharing can be achieved between vehicles and RSUs using dedicated short range communications (DSRC) [5] protocol.

As a new trend in future, although VANET will enhance travel safety and comfort for people's life, security issues should also be fully considered. In 2015, two network security engineers remotely intruded a Cherokee (free light) [5] on the road and performed deceleration, engine shutdown and out of control operations. In 2017, Tencent Keen Security Lab successfully cracked Tesla's autopilot system [6] twice, these researchers could open/close doors and control lighting system. Therefore, it is significant to establish a secure and effective authentication mechanism to avoid malicious intrusions.

In the existing related researches [7–9], a central authority in two-layer structure is completely trusted, we call it as trusted authority (TA). However, this authority may be compromised in real-world scenarios and leak sensitive information to third party without being legally authenticated. Especially, the TA may disclose registered users' private keys. This attack is denoted as registration authority leaks registration information (RALRI) attack. Generally, the leakage of private key will cause seriously traffic security problem due to the false messages (i.e. wrong predictions) generated by illegal user. Secondly, participants usually local calculate and cache sensitive data in advance to improve communication efficiency, where the random numbers can be used to establish session keys in mutual authentication phase. While these random numbers can be obtained by attacker through a new type of attack called ephemeral-secret-leakage (ESL) attack [9]. Email is the most common way to implement such attacks. Thirdly, the validity period of private keys for RSU and OBU must be equal in update phase, which means the synchronous update must be performed between different entities. Obviously, this strong correlation will degrade the flexibility of VANET system.

© The Institution of Engineering and Technology 2019

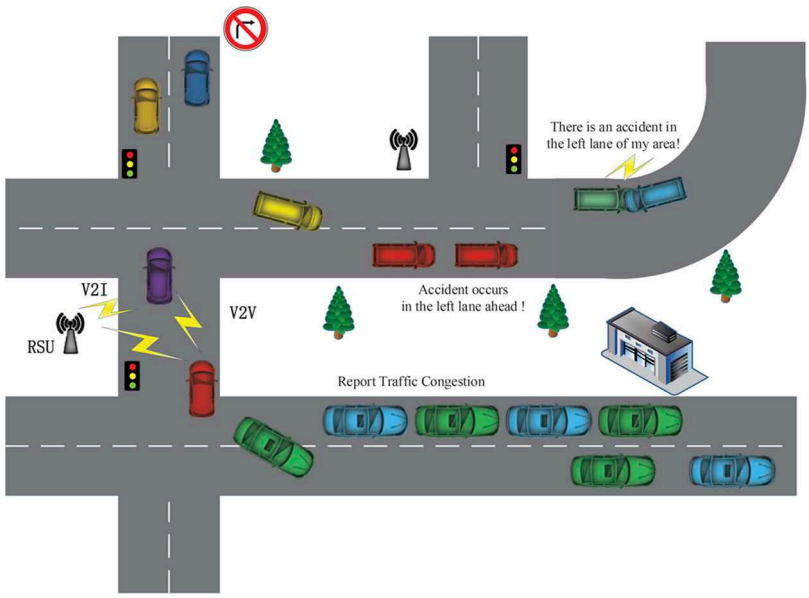

Fig. 1 *Basic structure of VANETs*

Based on self-certified public keys (SCPK) and Schnorr signatures, this paper proposes an efficient provably secure authentication protocol in VANET. The protocol can resist the above attacks and update private keys independently between RSU and OBU. It is shown that the proposed protocol is very suitable for practical VANET applications as it provides additional security requirements along with the reduced computation cost.

1.1 Our contributions

The contributions of this paper are summarised as follows:

- We analyse the security flaws of Tseng *et al.*'s scheme and propose a hierarchical revocable authentication protocol in VANET using bilinear pairing. It enables multiple subordinate authorities to work independently and eases the computation burden of the original single authority.
- We enhance the flexibility of private key update through embedding different validity periods in time keys. The validity period depends on the computation and storage capacity of specific devices.
- Our protocol is provably secure under the computational Diffie-Hellman (CDH) assumption in the modified BR model. It is resistant to the well-known security attacks such as RALRI attack, ESL attack and so on.
- Performance analysis shows that the protocol is significantly superior to most of the authentication schemes. It demonstrates that our protocol is suitable for the practical VANET environment.

The rest of this paper is organised as follows. Section 2 introduces some background, including preliminary knowledge, security model, system model and design goals. Section 3 gives the detailed process of our proposed protocol. The modified adversary model and security proof are presented in Section 4. The performance analysis is presented in Sections 5. Conclusion is given in Section 6.

1.2 Related work

In VANET, authentication is a significant cryptographic process of trust establishment between vehicles and infrastructure. Well-designed authentication schemes can easily identify non-legitimate nodes and fake message and thereby provide security for VANET. So far, many authentication schemes have been proposed to provide secure communication. Depending on the research in literatures [10–12], cryptography authentication schemes are divide into three categories: (i) asymmetric cryptography (public key infrastructure, PKI), (ii) symmetric cryptography (hash function) and (iii) identity-based cryptography (IBC) schemes.

Asymmetric cryptography (also called the public key infrastructure) is the most popular method to provide data security for communication networks. A pair of public/private keys can be used to encrypt a message or generate a digital signature. More specifically, PKI schemes depend on a trusted third party to authenticate vehicles and infrastructure. Wasef and Shen [13] proposed an expedite message authentication protocol scheme which adopts PKI. It is known that CRLs are used to perform revocation checking process and is very time consuming. This scheme overcomes this problem by replacing the CRL checking process with hash message authentication code.

The second authentication technique is the symmetric cryptography. It means that the same private key and simple algorithm are used to encrypt and decrypt message, and thereby the symmetric cryptography is more faster than asymmetric cryptography. Symmetric cryptography also has its limitations, such as it does not provide non-repudiation, and the leakage of private key could endanger all safety messages. Hash function based authentication scheme is one type of symmetric cryptography, and mainly used for checking the message integrity. In Chim *et al.*'s scheme [14], they used one-way hash function and secret key between vehicle and RSU to solve the problem of communication overhead. Vighnesh *et al.*'s scheme [15] utilised hash chain and authentication code to authenticate vehicles which send message. This authentication code is attached to each safety message, and encrypted by the receiving vehicle.

IBC-based authentication scheme is similar to asymmetric cryptography except that public key of a user in IBC scheme is easily derived from its personal information such as an email address. Compared with asymmetric cryptography, IBC system does not use certificates for message authentication and thus improves efficiency in VANET communication. In [16], Huang *et al.* proposed a batch authenticated scheme which significantly reduced computation cost. In [17], Sun *et al.*'s scheme used identity-based encryption for encryption, authentication and non-repudiation.

Privacy protection of vehicle user should also be taken into account. This is because that user's personal data may be intercepted by malicious party during the information exchange. In [18], Lin *et al.* proposed a conditional privacy-preserving protocol for VANETs. Their protocol uses short group signature to sign messages sent by vehicle and ensures the anonymity of signer. However, the verification of group signature is usually time consuming and not suitable to the high-mobility feature of VANET. Pseudonym-based authentication scheme can achieve conditional privacy preserving by frequently changing pseudonyms for vehicle

© The Institution of Engineering and Technology 2019

17518717, 2020, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-ifs.2019.0249, Wiley Online Library on [01/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

users. Literatures in [19, 20] used this approach to provide conditional privacy and traceability for malicious users.

2 Background

In this section, we compendiously introduce preliminary knowledge, security model, system model and design goals in this paper. Some necessary notations are listed in Nomenclature section.

2.1 Preliminary knowledge

2.1.1 Bilinear pairings: Let G_1 and G_2 be additive and multiplicative cyclic groups of large prime order q, respectively. A map $\hat{e}: G_1 \times G_1 \to G_2$ is called an admissible bilinear map if it satisfies the following three conditions:

i. *bilinearity*: for all $P, Q \in G_1$ and $a, b \in Z_q^*$, we have $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$;
ii. *non-degeneracy*: for some $P, Q \in G_1, \hat{e}(P, Q) \neq 1$ holds;
iii. *computability*: given $P, Q \in G_1$, there is an efficient algorithm to compute $\hat{e}(P, Q)$.

Note that condition (1) implies that: $\hat{e}(P + Q, R) = \hat{e}(P, R) \cdot \hat{e}(Q, R)$, and $\hat{e}(P, Q + R) = \hat{e}(P, Q) \cdot \hat{e}(P, R)$, for all $P, Q, R \in G_1$.

2.1.2 Security assumptions: Let G_1 and G_2 be defined as above. In this part, we define security assumptions on which our scheme is based.

- CDH problem: Given P, aP, $bP \in G_1$ for unknown $a, b \in Z_q^*$, the CDH problem in G_1 is to compute abP.
- CDH assumption: Given $P, aP, bP \in G_1$ for unknown $a, b \in Z_q^*$, no probabilistic polynomial time (PPT) adversary A can compute abP with a non-negligible probability ε.
- Adversary advantage: The successful probability of adversary A is presented below:

$$\mathrm{Adv}_A = \Pr\left[A(P, aP, bP) = abP \middle| P \in G_1, a, b \in Z_q^*\right], \quad (1)$$

where the probability is measured over the random choices of $a, b \in Z_q^*$ by adversary A.

2.2 BR93 model

The computational complexity analysis utilises deductive reasoning method. Although the method is proposed in 1980s, it became prevalent by Bellare and Rogaway (1993) [21] for key establishment protocols. In their literature, the first formal definition for adversary capability model was provided for two-party entity authentication protocols. The adversary is powerful to learn some sensitive information from interacting communication. Therefore, proving the indistinguishability between session key and same random key distribution for the adversary is important.

Informally, adversary can fully control communication network by interacting with a series of *oracles*. Each *oracle* represents a principal, who has an identity identifier U and oracle $\prod_U^s$. Specifically, adversary A intercepts all messages transmitted between two parties by interacting with oracles $\prod_{U_1,U_2}^i$, it represents the ith instance that principal U_1 wants to establish a session key with principal U_2. In the BR93 model, adversary can adaptively perform the following oracle queries:

- Send (U_1, U_2, i, m): This query means that adversary impersonates principal U_1 for sending message m to principal U_2, while the corresponding content is returned according to protocol specification. This query utilises the oracle $\prod_{U_1,U_2}^i$. If the received message m is not of the expected format, the oracle $\prod_{U_1,U_2}^i$ simply halt. Note that this query can also initiate a new protocol by sending an empty message m.
- *Reveal* (U_1, U_2, i)*:* This query means that oracle $\prod_{U_1,U_2}^i$ returns previously established session key, while returning a *null* value if it did not accept any session keys. It models the ability of adversary finding session keys, in the sense that a compromised session key should not endanger other session keys, i.e. the known-session key security. Note that oracle can only accept a session key once (while principal can accept many session keys in different oracles).
- *Corrupt* (U_1, K_E)*:* This query allows adversary to corrupt principal U_1 and learn its completely internal state at will. Especially, this adversary can replace the long-term secret key (i.e. private key K_E) of U_1 with its choice.
- *Test* (U_1, U_2, i)*:* This query means that adversary attempts to distinguish an accepted session key K_s by oracle $\prod_{U_1,U_2}^i$ from a random key with same distribution. This is the basis of determining security for protocol. Specifically, adversary chooses a randomly bit b. If $b = 0$, then K_s is returned while if $b = 1$, a random string is returned from the same distribution.

2.2.1 Definition of security: We define $\mathrm{SID}(\prod_{U_1,U_2}^i)$ as the concatenation of all messages related to oracle $\prod_{U_1,U_2}^i$, and denote $\mathrm{PID}(\prod_{U_1,U_2}^i)$ as the perceived partner of oracle $\prod_{U_1,U_2}^i$.

The security of BR93 model and its variants depend on the concept of *partnership* oracle, which is defined by different models. The BR93 model defines partnership using the notion of matching conversations.

Definition 1: (*Partnership):* Two oracles, $\prod_{U_1,U_2}^i$ and $\prod_{U_2,U_1}^j$, are partners, if both of the following requirements are satisfied:

(i) each one believes that the other is its partner (i.e. $\mathrm{PID}(\prod_{U_1,U_2}^i) = U_2$ and $\mathrm{PID}(\prod_{U_2,U_1}^j) = U_1$),
(ii) they agree on the same session identifier (i.e. $\mathrm{SID}(\prod_{U_1,U_2}^i) = \mathrm{SID}(\prod_{U_2,U_1}^j)$).

Definition 2: (Freshness): Oracle $\prod_{U_1,U_2}^i$ is fresh at the end of its execution, if and only if:

(i) Both oracles $\prod_{U_1,U_2}^i$ and $\prod_{U_2,U_1}^j$ have not been asked any *Reveal* queries,
(ii) both principals U_1 and U_2 have not been asked any *Corrupt* queries.

Security for protocol in BR93 model is defined by the following game G played between adversary and a series of oracles.

- *Stage 1:* Firstly, adversary A sends any oracle queries, including *Send*, *Reveal* and *Corrupt* query.
- *Stage 2:* Secondly, at some point in the game G, adversary A chooses a fresh session and sends a *Test* query to the associated oracle. In terms of the randomly chosen bit b, the adversary is given either actual session key or random key drawn from same distribution.
- *Stage 3:* Next, adversary continues querying oracles excepting for *Corrupt* and *Reveal* query that will eventually expose the tested session key.
- *Stage 4:* Finally, adversary terminates this game G and outputs a guess value b' of b.

In this game G, success for adversary is measured by its advantage in distinguishing the tested session key from a random key, i.e. the probability in outputting $b' = b$, then the advantage function for adversary A is presented as $\mathrm{Adv}^A(k) = \left|2\Pr[\mathrm{success}] - 1\right|$.

© The Institution of Engineering and Technology 2019

17518717, 2020, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-ifs.2019.0249, Wiley Online Library on [01/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

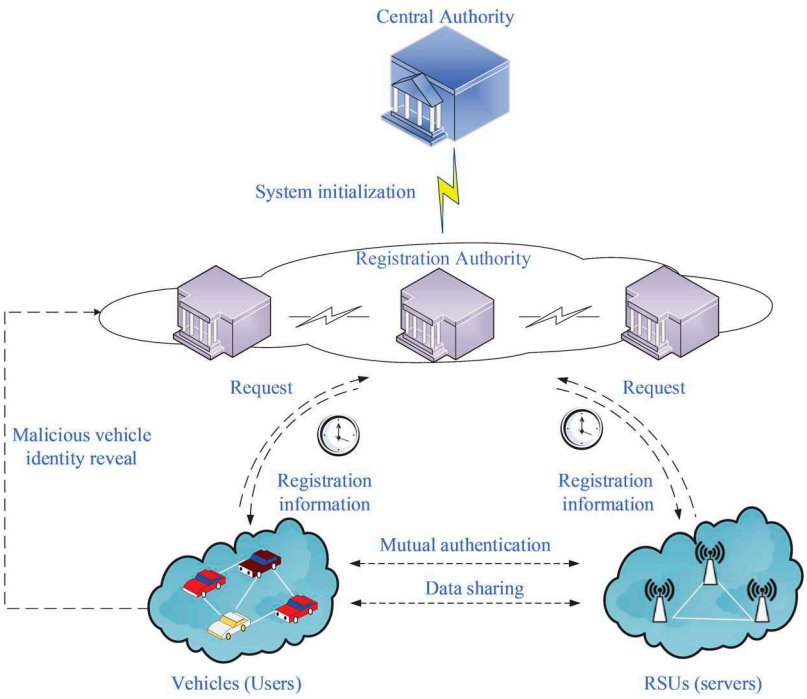

Fig. 2 *System model*

Definition 3: (BR93 Security): Secure protocol based on BR93 model should satisfy the following requirements:

(i) if protocol runs between two oracles $\prod^{i}_{U_1,U_2}$ and $\prod^{j}_{U_2,U_1}$ in the absence of malicious adversary, then both $\prod^{i}_{U_1,U_2}$ and $\prod^{j}_{U_2,U_1}$ can accept same session key,
(ii) for any PPT adversary A, the advantage $\mathrm{Adv}^{A}(k)$ is negligible.

2.3 System model

The system model of our protocol is given in Fig. 2. It consists of four entities: single central authority (CA), multiple registration authorities, RSUs and OBUs. The specific descriptions are as follows:

- *Central authority (*CA_1*):* The single authority at the top layer of system model. As the primary traffic administration bureau, the CA_1 is responsible for initialising the system parameters and assigning private keys for all the subordinate registration authorities.
- *Subordinate registration authority (*RA_2*):* The second-layer authority subject to central authority CA_1. As the regional traffic management apartments, the authority generates private keys and time keys for all the registered vehicle users and RSUs. It also revokes malicious users once detecting illegal behaviours.
- *RSUs:* Fixed infrastructures located aside roads. RSUs collect and share messages in networks with vehicles using DSRC.
- *OBUs:* Installed on vehicles to compute and store necessary values exchanging with RSUs. Note that these devices have limited computation and storage resources, so they cannot perform complex operations.

2.4 Design goals

In order to achieve secure communication in VANET network, we hope that our protocol could satisfy the following security requirements:

- *Mutual authentication:* Mutual authentication must be satisfied between vehicle users and RSUs to identify legitimate participants, and prevent malicious parties from participating in communication process.
- *Session key agreement:* To guarantee information security, our protocol should establish secure session key to encrypt the exchanged message.
- *User anonymity:* The real identity of vehicle users must be associated with personal information, such as ID card, home address, financial status and so on. Therefore, we need to provide user anonymity to avoid the leakage of sensitive data.
- *Conditional privacy:* Malicious users may accidently cause severe traffic accidents and criminal behaviours. Therefore, the system should recover the real identity to take necessary punishments for malicious users.
- *Flexibility of private key update:* The private key should be updated flexibly for RSU and OBU according to their respective computing and storage capabilities, thus saving computation resources.
- *Perfect forward secrecy:* Our protocol should provide perfect forward secrecy. That is, even though the private key are comprised in some way, the established session key is still secure.
- *Resistance of various attacks:* Our protocol should resist the well-known attacks, including that replay attack, man-in-the-middle attack, ESL attack and so on.

3 Our protocol

In this section, we present a new hierarchical revocable authentication protocol in VANET using bilinear pairing. Our protocol includes system initialisation, registration, time key update and revocation, and mutual authentication and key agreement. The detailed description is as follows.

3.1 System initialisation phase

In this phase, the single central authority and all the registration authorities execute the system initialisation.

3.1.1 Central authority (CA_1*) initialisation:* The central authority CA_1 selects same parameters as Section 2.1.1, and performs the following steps:

(i) Choose a random number $s \in Z_q^*$ as the system master key and compute $P_{\mathrm{pub}} = s \cdot P$ as the system public key;
(ii) Choose a map-to-point hash function and eight general one-way hash functions as follows:

© The Institution of Engineering and Technology 2019

17518717, 2020, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-ifs.2019.0249, Wiley Online Library on [01/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

$$H_1: \{0,1\}^* \to G_1, \quad H_i: \{0,1\}^* \to Z_q^*, \quad (i = 2, 3, \ldots, 9);$$

(iii) Publish the system parameters

$$\text{params} = \left\langle G_1, G_2, \hat{e}, q, P, P_{\text{pub}}, H_1, H_i (i = 2, 3, \ldots, 9) \right\rangle.$$

3.1.2 Registration authority (RA_2) initialisation: The central authorities CA_1 assigns a pair of private/public key for each registration authority RA_2. The detailed procedures are depicted as follows:

(i) The registration authority RA_2 sends its unique identifier RA_{ID_2} to the central authority CA_1;
(ii) After receiving it, the CA_1 generates the following private/public key pair for RA_2 and transfers private key d_{RA_2} to the RA_2 through secure channel

$$d_{RA_2} = s \cdot H_1(RA_{ID_2})$$
$$P_{RA_2} = H_2(d_{RA_2}) \cdot P$$

(iii) The authority RA_2 checks whether the equation $\hat{e}(d_{RA_2}, P) = \hat{e}(P_{\text{pub}}, H_1(RA_{ID_2}))$ holds or not. If holds, then the authority generates the ultimate private key $d_{URA_2} = H_2(d_{RA_2})$. Otherwise, it rejects the private key d_{RA_2}.

3.2 Registration phase

In this phase, vehicle user (OBU) and RSU register to authority RA_2 to get their private keys. Without loss of generality, we assume that RSU with identity ID_{sj} and vehicle user with ID_{ui} register to authority RA_{2k} and RA_{2e}, respectively. The simplified process is depicted in Fig. 3.

3.2.1 Registration phase for common RSU:

(i) The RSU selects an ephemeral secret $s_{sj} \in Z_q^*$, computes $S_{sj} = s_{sj} \cdot P$ and transfers $\langle ID_{sj}, S_{sj} \rangle$ to authority RA_{2k};
(ii) The authority RA_{2k} chooses an ephemeral secret $y_{sj} \in Z_q^*$, computes $Y_{sj} = y_{sj} \cdot P$ and generates private key with the format $d_{sj} = (y_{sj} + H_3(ID_{sj}, Y_{sj} + S_{sj}) \cdot d_{URA_{2k}}) \mod q$. Then, the RA_{2k} returns $\langle d_{sj}, Y_{sj}, RA_{ID_{2k}} \rangle$ to RSU via secure channel;
(iii) After receiving it, RSU checks whether the equation $d_{sj} \cdot P = Y_{sj} + H_3(ID_{sj}, Y_{sj} + S_{sj}) \cdot P_{RA_{2k}}$ holds or not. If holds, RSU accepts the private key d_{sj}. Otherwise, RSU rejects the message tuple.

3.2.2 Registration phase for anonymous vehicle: Vehicle user exploits pseudonym identity PID to protect its personal information. Then, the authority TA_{2e} verifies the legitimacy of vehicle user and generates private key for it.

(i) The vehicle selects a random number $a \in Z_q^*$ and an ephemeral secret $s_{ui} \in Z_q^*$, and computes $S_{ui} = s_{ui} \cdot P, F_{ui} = H_4(ID_{ui}) \cdot P + P_{RA_{2e}}, A_{ui} = (a + s_{ui}) \cdot P, PID_{ui}$. $= ID_{ui} \oplus H_5(A_{ui}), B_{ui} = (a + s_{ui}) \cdot F_{ui}$
Then, vehicle sends $\langle PID_{ui}, H_4(ID_{ui}), B_{ui}, S_{ui} \rangle$ to the authority RA_{2e};
(ii) After receiving it, the authority RA_{2e} computes $A_{ui} = (d_{URA_{2e}} + H_4(ID_{ui}))^{-1} \cdot B_{ui}$. The real identity of vehicle user could be obtained by $ID_{ui} = PID_{ui} \oplus H_5(A_{ui})$;
(iii) For legal vehicle user, the authority RA_{2e} selects an ephemeral secret $y_{ui} \in Z_q^*$ and computes $Y_{ui} = y_{ui} \cdot P$, then returns private key $d_{ui} = (y_{ui} + H_3(PID_{ui}, Y_{ui} + S_{ui}) \cdot d_{URA_{2e}}) \mod q$ to vehicle via secure channel.

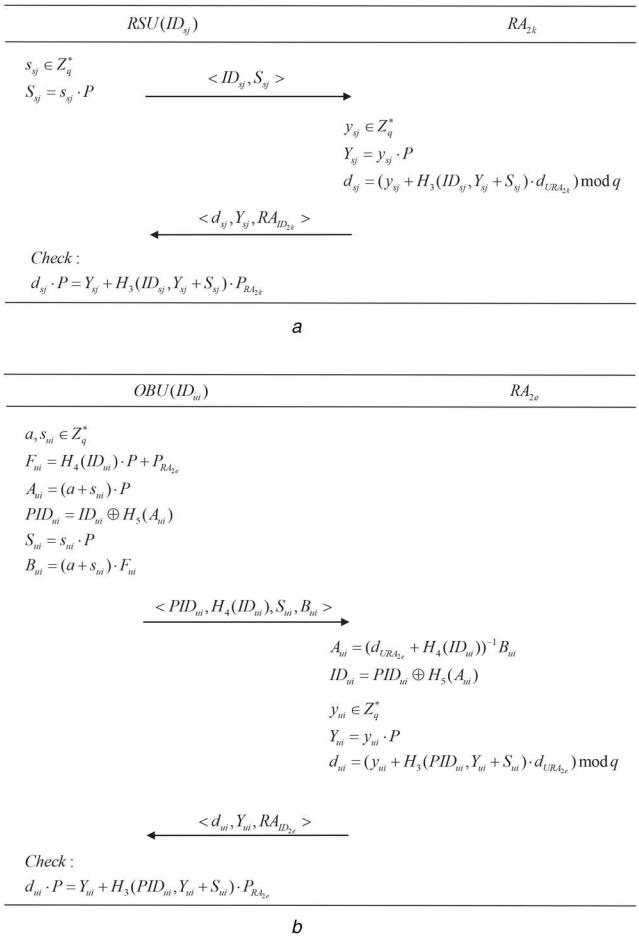

Fig. 3 *Registration phase*
(a) Registration phase for common RSU, ***(b)*** Registration phase for anonymous vehicle

© The Institution of Engineering and Technology 2019

17518717, 2020, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-ifs.2019.0249, Wiley Online Library on [01/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

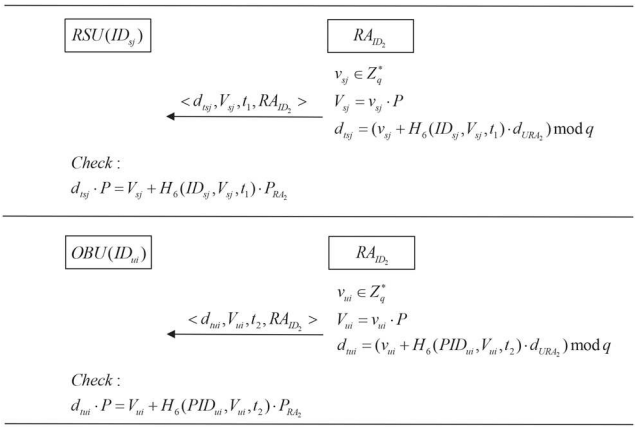

Fig. 4 *Time key update and revocation*

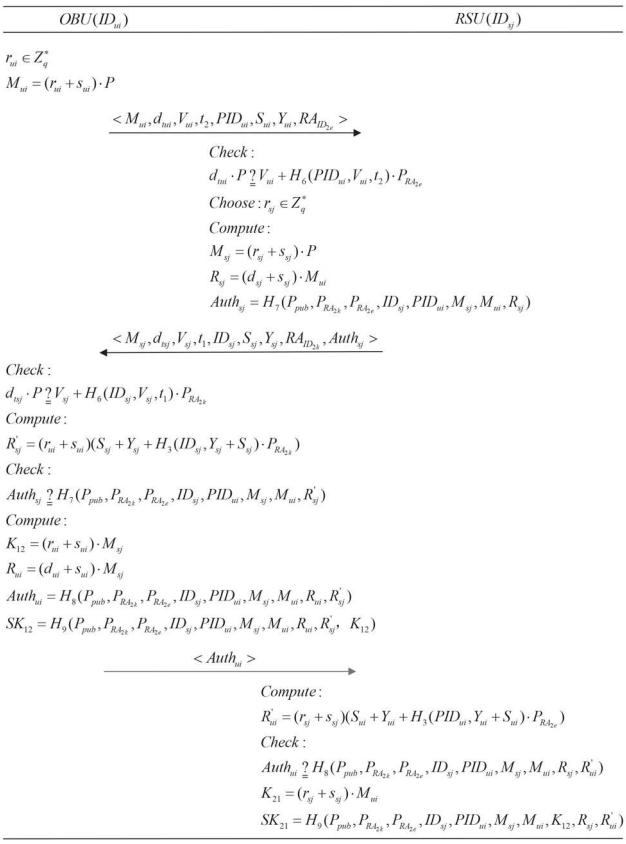

Fig. 5 *Mutual authentication and key agreement phase*

(iv) Upon receiving $\langle d_{ui}, Y_{ui}, \mathrm{RA}_{\mathrm{ID}_{2e}} \rangle$, vehicle checks whether the equation $d_{ui} \cdot P = Y_{ui} + H_3(\mathrm{PID}_{ui}, Y_{ui} + S_{ui}) \cdot P_{\mathrm{RA}_{2e}}$ holds or not. If holds, the vehicle accepts the private key d_{ui}. Otherwise, the vehicle user rejects the message.

3.3 Time key update and revocation phase

In order to assign different validity periods for private keys, we embed different time in time keys to ensure flexible update. The time is selected based on actual computation and storage capability. This process is presented in Fig. 4:

(i) The authority RA_{2k} selects validity period t_1 for RSU;

(ii) The RA_{2k} selects ephemeral secret $v_{sj} \in Z_q^*$ and computes $V_{sj} = v_{sj} \cdot P$. Then, RA_{2k} generates the time key $d_{tsj} = (v_{sj} + H_6(\mathrm{ID}_{sj}, V_{sj}, t_1) \cdot d_{\mathrm{URA}_{2k}})(\bmod q)$, and sends tuple $\langle d_{tsj}, V_{sj}, t_1, \mathrm{RA}_{\mathrm{ID}_{2k}} \rangle$ to RSU;

(iii) After receiving it, RSU verifies the time key by the equation $d_{tsj} \cdot P \stackrel{?}{=} V_{sj} + H_6(\mathrm{ID}_{sj}, V_{sj}, t_1) \cdot P_{\mathrm{RA}_{2k}}$.

The generation of time key for vehicle is similar to the above process for RSU:

(i) The authority RA_{2e} selects validity period t_2 for vehicle;

(ii) The RA_{2e} selects ephemeral secret $v_{ui} \in Z_q^*$ and computes $V_{ui} = v_{ui} \cdot P$. Then, RA_{2e} generates time key $d_{tui} = (v_{ui} + H_6(\mathrm{PID}_{ui}, V_{ui}, t_2) \cdot d_{\mathrm{URA}_{2e}})(\bmod q)$, and sends the tuple $\langle d_{tui}, V_{ui}, t_2, \mathrm{RA}_{\mathrm{ID}_{2e}} \rangle$ to vehicle;

(iii) After receiving it, vehicle verifies the time key by the equation $d_{tui} \cdot P \stackrel{?}{=} V_{ui} + H_6(\mathrm{PID}_{ui}, V_{ui}, t_2)) \cdot P_{\mathrm{RA}_{2e}}$.

Note that, if the equation does not hold, then the authority needs to update corresponding private key.

3.4 Mutual authentication and key agreement phase

In this section, the mutual authentication and key agreement is performed between vehicle and RSU without involvement of any authority. This phase includes three handshakes and specific procedure is given in Fig. 5.

Firstly, vehicle user sends request information with its anonymous identity PID_{ui}:

(i) Choose an ephemeral secret $r_{ui} \in Z_q^*$, and compute $M_{ui} = (r_{ui} + s_{ui}) \cdot P$;

(ii) Send $\langle M_{ui}, d_{tui}, V_{ui}, t_2, \mathrm{PID}_{ui}, S_{ui}, Y_{ui}, \mathrm{RA}_{\mathrm{ID}_{2e}} \rangle$ as the request message to RSU.

Secondly, RSU performs the following procedures:

(i) Check the public key $P_{\mathrm{RA}_{2e}}$ of authority RA_{2e} whose vehicle registers with, and check the time key by equation $d_{\mathrm{tui}} \cdot P = V_{ui} + H_6(\mathrm{PID}_{ui}, V_{ui}, t_2) \cdot P_{\mathrm{RA}_{2e}}$. If holds, the time key d_{tui} for vehicle is valid. If the public key $P_{\mathrm{RA}_{2e}}$ does not found or the equation does not hold, RSU terminates this session. That is because the authority RA_{2e} or vehicle has been revoked;

(ii) Choose an ephemeral secret $r_{sj} \in Z_q^*$, compute $M_{sj} = (r_{sj} + s_{sj}) \cdot P$, $R_{sj} = (d_{sj} + s_{sj}) \cdot M_{ui}$ and its certificate $\mathrm{Auth}_{sj} = H_7(P_{\mathrm{pub}}, P_{\mathrm{RA}_{2k}}, P_{\mathrm{RA}_{2e}}, \mathrm{ID}_{sj}, \mathrm{PID}_{ui}, M_{sj}, M_{ui}, R_{sj})$;

(iii) Return $\langle M_{sj}, d_{tsj}, V_{sj}, t_1, \mathrm{ID}_{sj}, S_{sj}, Y_{sj}, \mathrm{RA}_{\mathrm{ID}_{2k}}, \mathrm{Auth}_{sj} \rangle$ to the vehicle.

Next, vehicle user verifies RSU and computes a session key:

(i) Check the public key $P_{\mathrm{RA}_{2k}}$ of authority RA_{2k} whose RSU registers with, and check the time key by equation $d_{tsj} \cdot P = V_{sj} + H_6(\mathrm{ID}_{sj}, V_{sj}, t_1) \cdot P_{\mathrm{URA}_{2k}}$. If holds, the time key d_{tsj} for RSU is valid. If the public key $P_{\mathrm{RA}_{2k}}$ does not found or the equation does not hold, vehicle terminates this session. That is because the authority RA_{2k} or RSU has been revoked;

© The Institution of Engineering and Technology 2019

17518717, 2020, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-ifs.2019.0249, Wiley Online Library on [01/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

(ii) Compute $R'_{sj} = (r_{ui} + s_{ui})(S_{sj} + Y_{sj} + H_3(\mathrm{ID}_{sj}, Y_{sj} + S_{sj}) \cdot P_{\mathrm{RA}_{2k}})$ and verify certificate Auth_{sj} according to $\mathrm{Auth}'_{sj} \stackrel{?}{=} H_7(P_{\mathrm{pub}}, P_{\mathrm{RA}_{2k}}, P_{\mathrm{RA}_{2e}}, \mathrm{ID}_{sj}, \mathrm{PID}_{ui}, M_{sj}, M_{ui}, R'_{sj})$;
(iii) If the above equation holds, vehicle computes $K_{12} = (r_{ui} + s_{ui}) \cdot M_{sj}$, $R_{ui} = (d_{ui} + s_{ui}) \cdot M_{sj}$, certificate $\mathrm{Auth}_{ui} = H_8(P_{\mathrm{pub}}, P_{\mathrm{RA}_{2k}}, P_{\mathrm{RA}_{2e}}, \mathrm{ID}_{sj}, \mathrm{PID}_{ui}, M_{sj}, M_{ui}, R_{sj}, R'_{sj})$, and session key $\mathrm{SK}_{12} = H_9(P_{\mathrm{pub}}, P_{\mathrm{RA}_{2k}}, P_{\mathrm{RA}_{2e}}, \mathrm{ID}_{sj}, \mathrm{PID}_{ui}, M_{sj}, M_{ui}, R_{ui}, R'_{sj}, K_{12})$;
(iv) Return its certification $\langle \mathrm{Auth}_{ui} \rangle$ to the RSU.

Finally, RSU agrees on a session key with vehicle by the following steps:

(i) Compute $R'_{ui} = (r_{sj} + s_{sj})(S_{ui} + Y_{ui} + H_3(\mathrm{ID}_{ui}, Y_{ui} + S_{ui}) \cdot P_{\mathrm{RA}_{2e}})$;
(ii) Check the equation $\mathrm{Auth}'_{ui} \stackrel{?}{=} H_8(P_{\mathrm{pub}}, P_{\mathrm{RA}_{2k}}, P_{\mathrm{RA}_{2e}}, \mathrm{ID}_{sj}, \mathrm{PID}_{ui}, M_{sj}, M_{ui}, R_{sj}, R'_{ui})$ holds or not. If bolds, it turns to next step; otherwise, this session is terminated;
(iii) Compute $K_{21} = (r_{sj} + s_{sj}) \cdot M_{ui}$ and agree on the session key $\mathrm{SK}_{21} = H_9(P_{\mathrm{pub}}, P_{\mathrm{RA}_{2k}}, P_{\mathrm{RA}_{2e}}, \mathrm{ID}_{sj}, \mathrm{PID}_{ui}, M_{sj}, M_{ui}, R_{sj}, R'_{ui}, K_{21})$.

During the mutual authentication and key agreement phase, legal entity needs to ensure that $R_{sj} = R'_{sj}$ and $R_{ui} = R'_{ui}$. The specific correctness analysis is as follows:

$$\begin{aligned} R_{sj} &= (d_{sj} + s_{sj}) \cdot M_{ui} \\ &= (d_{sj} + s_{sj}) \cdot (r_{ui} + s_{ui}) \cdot P \\ &= (r_{ui} + s_{ui}) \cdot (d_{sj} + s_{sj}) \cdot P \\ &= (r_{ui} + s_{ui}) \cdot (y_{sj} + H_3(\mathrm{ID}_{sj}, Y_{sj} + S_{sj}) \cdot d_{URA_{2k}} + s_{sj}) \cdot P \\ &= (r_{ui} + s_{ui}) \cdot (Y_{sj} + S_{sj} + H_3(\mathrm{ID}_{sj}, Y_{sj} + S_{sj}) \cdot P_{TA_{2k}}) \\ &= R'_{sj}. \end{aligned} \tag{2}$$

$$\begin{aligned} R_{ui} &= (d_{ui} + s_{ui}) \cdot M_{sj} \\ &= (d_{ui} + s_{ui}) \cdot (r_{sj} + s_{sj}) \cdot P \\ &= (r_{sj} + s_{sj}) \cdot (d_{ui} + s_{ui}) \cdot P \\ &= (r_{sj} + s_{sj}) \cdot (y_{ui} + H_3(\mathrm{ID}_{ui}, Y_{ui} + S_{ui}) \cdot d_{URA_{2e}} + s_{ui}) \cdot P \\ &= (r_{sj} + s_{sj}) \cdot (Y_{ui} + S_{ui} + H_3(\mathrm{ID}_{ui}, Y_{ui} + S_{ui}) \cdot P_{TA_{2e}}) \\ &= R'_{ui}. \end{aligned} \tag{3}$$

4 Security analysis

4.1 Modified BR93 model

In this section, we improve the BR93 model by adding a series of new oracle queries. These oracles allow adversary to arbitrarily access the information he needs and thereby enhance its attack capacity. Except for the *Send*, *Reveal*, *Corrupt* and *Test* query same as the BR93 model, we define new oracle queries as follows:

- $Hash_i(M)$ query: This oracle maintains an initial empty list for each hash function. It responds to adversary with specific message according to the received M. Upon receiving this query, the same response is given if the query has been asked before. Otherwise, the oracle randomly selects a value $r_i \in Z_q^*$ and records (M, r_i) in the list L_i, while returning r_i to adversary A.
- *Extract(U)* query: Upon receiving this query, the same response is given if the query has been asked before. Otherwise, the oracle computes private key d_U of U and returns to adversary. The difference between *Extract* and *Corrupt* query is that *Extract* query is sent before a session is established, while *Corrupt* query is sent after the session key agreement is completed.
- *Update* (U, t) query: This query allows adversary to obtain time key. Upon receiving this query, the oracle generates time key d_{tU} associated with (U, t) and sends to adversary.
- ESL(U) query: Upon receiving this query, the oracle returns random numbers used by instance U to adversary. This query models the ephemeral secret leakage (ESL) attacks.

For convenience, *client* and *server* represent the vehicle user and RSU, respectively, to illustrate the mutual authentication process. Next, we describe some definitions [22] related to security properties.

Definition 4: (Client-to-server authentication): Client-to-server authentication indicates that an adversary without the private key of a legitimate client cannot impersonate the client to communicate with server.

Definition 5: (Server-to-client authentication): Server-to-client authentication means that an adversary without the private key of a legitimate server cannot impersonate the server to communicate with client.

Definition 6: (Implicit key confirmation): Implicit key confirmation means that protocol assures that only the server/client can compute a session key that no others can produce.

Definition 7: (Full forward secrecy): If an adversary cannot recover previous established session keys after corrupting both private keys of client and server, we say that the protocol offers full forward secrecy.

4.2 Provable security

In this section, we will show that our protocol is provably secure in the modified BR93 model under the CDH assumption. Therefore, we need to construct an algorithm (simulation) B to solve CDH problem. Namely, given a random instance $\langle P, aP, bP \rangle$ in G_1 with unknown $a, b \in Z_q^*$, the algorithm B attempts to derive abP by interacting with adversary.

We instantiate a CDH problem $\{P, Q_1 = x \cdot P, Q_2 = y \cdot P\}$. Let q_c, q_s, q_{ns} and q_i denote the number of clients, servers, sessions and hash query H_i, respectively. Simulation B randomly picks $S' \in \{q_1, q_2 \ldots, q_s\}$, $C' \in \{q_1, q_2 \ldots, q_c\}$, and $x, y \in \{q_1, q_2 \ldots, q_{ns}\}$. We use the notations $\prod_C^i$ and $\prod_S^j$ to indicate the ith instance of client (vehicle user) and the jth instance of server (RSU), respectively. Meanwhile, we assume that client instance $\prod_{C'}^x$ and server instance $\prod_{S'}^y$ are partners.

The simulation B initialises the system parameters as follows: simulation B firstly generates system parameters $\mathrm{params} = \langle G_1, G_2, \hat{e}, q, P, P_{\mathrm{pub}}, H_1, H_i (i = 2, 3, \ldots, 9) \rangle$, where $P_{\mathrm{pub}} = z \cdot P$ as system public key for CA_1 and z is the corresponding private key. Then B generates $\langle \mathrm{RA}_{2k}, \mathrm{RA}_{\mathrm{ID}_{2k}}, P_{\mathrm{RA}_{2k}} = z_k \cdot P \rangle$ for registration authority RA_{2k}, where z and z_k are random values in Z_q^*. Finally, simulation B sends the public parameters 'params' and public key of authority RA_{2k} to adversary A. Note that simulation B picks nine random oracles as hash functions $H_i, i = 1, \ldots, 9$, and maintains nine lists L_i, for $i = 1, \ldots, 9$, which are initial empty list.

Next, we conceal the random instance $\{P, Q_1 = x \cdot P, Q_2 = y \cdot P\}$ of CDH problem inside the specific queries. The adversary can send eight kinds of queries, including *Hash*, *Extract*, *Update*, *Corrupt*, *ESL*, *Send*, *Reveal* and *Test* query. The response way of simulation B is as follows:

- • $H_i(M)$ query: The response to *Hash* query is the same as Section 4.1.
- *Extract* (ID_U) query: Simulation B maintains a list L_e about $(\mathrm{ID}_{sj}, Y_{sj}, S_{sj}, r_3, d_{sj}, \mathrm{RA}_{\mathrm{ID}_{2k}})$. Upon receiving $(\mathrm{ID}_{sj}, S_{sj})$, simulation B returns the same response to adversary if this query has been asked before. Specifically, B firstly chooses random numbers $a, b \in Z_q^*$, and computes $Y_{sj} = a \cdot P + b \cdot P_{RA_{2k}}$. Then, B computes $d_{sj} = a \mod q$, $r_3 = (-b) \bmod q$ and records

© The Institution of Engineering and Technology 2019

17518717, 2020, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-ifs.2019.0249, Wiley Online Library on [01/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

$(\mathrm{ID}_{sj}, Y_{sj}, S_{sj}, r_3)$ and $(\mathrm{ID}_{sj}, Y_{sj}, S_{sj}, r_3, d_{sj}, \mathrm{RA}_{\mathrm{ID}_{2k}})$ in L_3 and L_e, respectively. Finally, simulation B returns $(Y_{sj}, d_{sj}, \mathrm{RA}_{\mathrm{ID}_{2k}})$ to A.

- *Update* (ID, t) query: Simulation B maintains a list L_t about $(\mathrm{ID}_{sj}, V_{sj}, t_1, r_6, d_{tsj}, \mathrm{RA}_{\mathrm{ID}_{2k}})$. Upon receiving $(\mathrm{ID}_{sj}, t_1, \mathrm{RA}_{\mathrm{ID}_{2k}})$, simulation B returns the same response to adversary if the query has been asked before. Generally, B firstly chooses random numbers $a, b \in Z_q^*$ and computes $V_{sj} = a \cdot P + b \cdot P_{RA_{2k}}$. Then, B computes $d_{tsj} = a \bmod q$, $r_6 = (-b) \bmod q$ and records $(\mathrm{ID}_{sj}, V_{sj}, t_1, r_6)$ and $(\mathrm{ID}_{sj}, V_{sj}, t_1, r_6, d_{tsj}, \mathrm{RA}_{\mathrm{ID}_{2k}})$ in L_6 and L_t, respectively. Finally, simulation B returns the latter to adversary.
- *Corrupt* $(\prod_U^k)$ query: Upon receiving this query, simulation B returns both private key and time key of instance U to adversary.
- ESL$(\prod_U^k)$: The response to ESL query is the same as Section 4.1.
- Send$(\prod_U^k, M)$: There are three cases according to the received message M.

Case 1: $M =$' begin'.

(i) If $\prod_C^i \neq \prod_{C'}^x$, B selects a random value $r_{ui} \in Z_q^*$ and computes $M_{ui} = (r_{ui} + s_{ui}) \cdot P$, then B returns $\langle M_{ui}, d_{tui}, V_{ui}, t_2, \mathrm{PID}_{ui}, S_{ui}, Y_{ui}, \mathrm{RA}_{\mathrm{ID}_{2e}} \rangle$ to A.

(ii) If $\prod_C^i = \prod_{C'}^x$, B selects a random value $a \in Z_q^*$ and computes $M_{ui} = a \cdot P + Q_1$, then B returns $\langle M_{ui}, d_{tui}, V_{ui}, t_2, \mathrm{PID}_{ui}, S_{ui}, Y_{ui}, \mathrm{RA}_{\mathrm{ID}_{2e}}, a \rangle$ to adversary A.

Case 2: $M = \langle M_{ui}, d_{tui}, V_{ui}, t_2, \mathrm{PID}_{ui}, S_{ui}, Y_{ui}, \mathrm{RA}_{\mathrm{ID}_{2e}} \rangle$.

(i) If $\prod_C^i \neq \prod_{C'}^x$ and $\prod_S^j \neq \prod_{S'}^y$, simulation B responds to adversary according to the protocol specification.

(ii) If $\prod_C^i \neq \prod_{C'}^x$ and $\prod_S^j = \prod_{S'}^y$, simulation B checks the authority RA_{2e} and time key d_{tui}. When both of them are valid, B selects a random value $c \in Z_q^*$, and computes $M_{sj} = cP + Q_2$, $R_{sj} = (r_{ui} + s_{ui})(Q_2 + Y_{sj} + H_3(\mathrm{ID}_{sj}, Y_{sj} + S_{sj}) \cdot P_{\mathrm{RA}_{2k}})$ and $\mathrm{Auth}_{sj} = H_7(P_{\mathrm{pub}}, P_{\mathrm{RA}_{2k}}, P_{\mathrm{RA}_{2e}}, \mathrm{ID}_{sj}, \mathrm{PID}_{ui}, M_{sj}, M_{ui}, R_{sj})$. Then, B returns $\langle M_{sj}, d_{tsj}, V_{sj}, t_1, \mathrm{ID}_{sj}, S_{sj}, Y_{sj}, \mathrm{RA}_{\mathrm{ID}_{2k}}, \mathrm{Auth}_{sj}, c \rangle$ to adversary. If the authority or time key is invalid, simulation B terminates this session.

(iii) If $\prod_C^i = \prod_{C'}^x$ and $\prod_S^j \neq \prod_{S'}^y$, B checks the authority RA_{2e} and time key d_{tui}. When both of them are valid, simulation B responds to adversary according to the protocol.

(iv) If $\prod_C^i = \prod_{C'}^x$ and $\prod_S^j = \prod_{S'}^y$, simulation B checks the validity of registration authority RA_{2e} and time key d_{tui}. When both of them are valid, B selects random numbers $c, d \in Z_q^*$ and computes $M_{sj} = cP + Q_2$, $S_{sj} = Q_2$, $R_{sj} = dP$ and $\mathrm{Auth}_{sj} = H_7(P_{\mathrm{pub}}, P_{\mathrm{RA}_{2k}}, P_{\mathrm{RA}_{2e}}, \mathrm{ID}_{sj}, \mathrm{PID}_{ui}, M_{sj}, M_{ui}, R_{sj})$. Then, B returns $\langle M_{ui}, d_{tui}, V_{ui}, t_2, \mathrm{PID}_{ui}, S_{ui}, Y_{ui}, \mathrm{TA}_{\mathrm{ID}_{2e}}, c \rangle$ to adversary.

Case 3: $M = \langle M_{sj}, d_{tsj}, V_{sj}, t_1, ID_{sj}, S_{sj}, Y_{sj}, \mathrm{RA}_{\mathrm{ID}_{2k}}, \mathrm{Auth}_{sj} \rangle$.

(i) If $\prod_C^i \neq \prod_{C'}^x$ and $\prod_S^j \neq \prod_{S'}^y$, simulation B responds to adversary according to the protocol.

(ii) If $\prod_C^i \neq \prod_{C'}^x$ and $\prod_S^j = \prod_{S'}^y$, simulation B checks the authority RA_{2k} and time key d_{tsj}. If both of them are valid, B computes $R'_{sj} = (r_{ui} + s_{ui})(Q_2 + Y_{sj} + H_3(\mathrm{ID}_{sj}, Y_{sj} + Q_2) \cdot P_{\mathrm{RA}_{2k}})$. Then, B checks the equation $\mathrm{Auth}'_{sj} \stackrel{?}{=} H_7(P_{\mathrm{pub}}, P_{\mathrm{RA}_{2k}}, P_{\mathrm{RA}_{2e}}, \mathrm{ID}_{sj}, \mathrm{PID}_{ui}, M_{sj}, M_{ui}, R'_{sj})$ holds or not. If holds, B computes $K_{12} = (r_{ui} + s_{ui}) \cdot M_{sj}$. Otherwise, simulation B terminates this session.

(iii) If $\prod_C^i = \prod_{C'}^x$ and $\prod_S^j \neq \prod_{S'}^y$, simulation B responds to adversary according to the protocol.

(iv) If $\prod_C^i = \prod_{C'}^x$ and $\prod_S^j = \prod_{S'}^y$, simulation B checks the authority RA_{2k} and time key d_{tsj}. If both of them are valid, B computes $R'_{sj} = R_{sj} = d \cdot P$ and checks the equation $\mathrm{Auth}'_{sj} \stackrel{?}{=} H_7(P_{\mathrm{pub}}, P_{\mathrm{RA}_{2k}}, P_{\mathrm{RA}_{2e}}, \mathrm{ID}_{sj}, \mathrm{PID}_{ui}, M_{sj}, M_{ui}, R'_{sj})$ holds or not. If holds, B chooses random values $b, e \in Z_q^*$ and computes $R_{ui} = bP, K_{12} = eM_{sj}$, $\mathrm{Auth}_{ui} = H_8(P_{\mathrm{pub}}, P_{\mathrm{RA}_{2k}}, P_{\mathrm{RA}_{2e}}, \mathrm{ID}_{sj}, \mathrm{PID}_{ui}, M_{sj}, M_{ui}, R_{ui}, R'_{sj})$, and session key $\mathrm{SK}_{12} = H_9(P_{\mathrm{pub}}, P_{\mathrm{RA}_{2k}}, P_{\mathrm{RA}_{2e}}, \mathrm{ID}_{sj}, \mathrm{PID}_{ui}, M_{sj}, M_{ui}, R_{ui}, R'_{sj}, K_{12})$. Finally, B returns $\langle \mathrm{Auth}_{ui} \rangle$ to adversary.

- Reveal$(\prod_U^k)$: There are four cases in terms of the instance $\prod_U^k$.

(i) If $\prod_C^i \neq \prod_{C'}^x$ and $\prod_S^j \neq \prod_{S'}^y$, B returns the established session key SK once the instance $\prod_U^k$ accepts a session. Otherwise, B returns a null value.

(ii) If $\prod_C^i \neq \prod_{C'}^x$ and $\prod_S^j = \prod_{S'}^y$, B returns $\mathrm{SK}_{12} = H_9(P_{\mathrm{pub}}, P_{\mathrm{RA}_{2k}}, P_{\mathrm{RA}_{2e}}, \mathrm{ID}_{sj}, \mathrm{PID}_{ui}, M_{sj}, M_{ui}, R_{ui}, R'_{sj}, K_{12})$ once the instance $\prod_U^k$ accepts a session, where $R'_{sj} = (r_{ui} + s_{ui})(Q_2 + Y_{sj} + H_3(\mathrm{ID}_{sj}, Y_{sj} + Q_2) \cdot P_{\mathrm{RA}_{2k}})$, $R_{ui} = (d_{ui} + s_{ui}) \cdot M_{sj}, K_{12} = (r_{ui} + s_{ui}) \cdot M_{sj}$.

(iii) If $\prod_C^i = \prod_{C'}^x$ and $\prod_S^j \neq \prod_{S'}^y$, S returns $\mathrm{SK}_{21} = H_9(P_{\mathrm{pub}}, P_{\mathrm{RA}_{2k}}, P_{\mathrm{RA}_{2e}}, \mathrm{ID}_{sj}, \mathrm{PID}_{ui}, M_{sj}, M_{ui}, R_{sj}, R'_{ui}, K_{12})$, where $R'_{ui} = (r_{sj} + s_{sj})(Q_1 + Y_{ui} + H_3(\mathrm{ID}_{ui}, Y_{ui} + Q_1) \cdot P_{\mathrm{RA}_{2e}})$, $R_{sj} = (d_{sj} + s_{sj}) \cdot M_{ui}, K_{12} = (r_{sj} + s_{sj}) \cdot M_{ui}$.

(iv) If $\prod_C^i = \prod_{C'}^x$ and $\prod_S^j = \prod_{S'}^y$, simulation B returns a null value.

In the following content, we prove the security properties of the protocol with five theorems. The security properties includes: (i) server-to-client authentication, (ii) client-to-server authentication, (iii) key agreement, (iv) implicit key confirmation and (v) full forward secrecy.

4.2.1 Server-to-client authentication: Let the symbol Event^{S2C} denote the event violating the server-to-client authentication when instance $\prod_C^i$ accepts an adversary who does not possess a legitimate private key of server. We denote the event probability as Pr [*S2C*].

Theorem 1: In the modified BR93 model, suppose that a PPT adversary can violate the server-to-client authentication with non-negligible advantage, then we can construct an algorithm to solve the CDH problem with non-negligible advantage.

Proof: We discuss the event probability Pr [*S2C*] of server-to-client authentication. The probability of event $\prod_C^i = \prod_{C'}^x$ and $\prod_S^j = \prod_{S'}^y$ happen simultaneously is $1/q_n q_m$. If adversary do not send ESL$(\prod_U^k)$ and Reveal$(\prod_U^k)$ query, algorithm B will not terminate this game. Meanwhile, the probability of achieving a successful session is $1/q_s^2$. As H_3 is a random oracle, B can find R_{sj} in the list L_3 in the probability of $1/q_i$. Therefore, we can compute xyP in the following equation:

$$xyP = R_{sj} - d_{sj}Q_1 - aQ_2 - ad_{sj}P \tag{4}$$

where d_{sj} can be obtained by Corrupt$(\prod_U^k)$ query.

Therefore, if an adversary can impersonate server to authenticate with client in a non-negligible advantage ε, then simulation B can solve CDH problem in a non-negligible advantage

© The Institution of Engineering and Technology 2019

17518717, 2020, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-ifs.2019.0249, Wiley Online Library on [01/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

$$\Pr[S2C] = \frac{\varepsilon}{q_n q_m q_s^2 q_i} < \varepsilon . \tag{5}$$

In summary, our protocol can achieve the server-to-client authentication. □

4.2.2 Client-to-server authentication: Let the symbol Event^{C2S} denote the event violating the client-to-server authentication when instance $\prod_S^j$ accepts an adversary who does not possess a legitimate private key of client. We denote the event probability as $\Pr[C2S]$.

Theorem 2: In the modified BR93 model, suppose that a PPT adversary can violate the client-to-server authentication with non-negligible advantage, then we can construct an algorithm B to solve the CDH problem with non-negligible advantage.

Proof: We discuss the event probability $\Pr[C2S]$ of client-to-server authentication. The proof process is similar to server-to-client authentication in Theorem 1. The probability of event $\prod_C^i = \prod_{C'}^x$ and $\prod_S^j = \prod_{S'}^y$ happen simultaneously is $1/q_n q_m$. Meanwhile, the probability of achieving a successful session is $1/q_s^2$. As H_3 is a random oracle, B can find R'_{sj} or R_{ui} in the list L_3 in the probability of $1/q_i$. Therefore, we can compute xyP in the following equation:

$$\begin{gathered} xyP = R'_{sj} - d_{sj}Q_1 - aQ_2 - ad_{sj}P \\ \text{or} \\ xyP = R_{sj} - cQ_1 - d_{ui}Q_2 - cd_{ui}P . \end{gathered} \tag{6}$$

where, d_{sj}, d_{ui} can be obtained by $\text{Corrupt}(\prod_U^k)$ query.

Therefore, if an adversary can impersonate client to authenticate with server in a non-negligible advantage ε, then simulation B can solve CDH problem in a non-negligible advantage

$$\Pr[C2S] = \frac{2\varepsilon}{q_n q_m q_s^2 q_i} < \varepsilon . \tag{7}$$

In summary, our protocol can achieve the client-to-server authentication. □

4.2.3 Key agreement: We show that our protocol provides key agreement in the modified BR93 model under the Diffie-Hellman assumption.

Theorem 3: In the modified BR93 model, assuming that adversary can guess the correct bit b involved in the $\text{Test}(\prod_U^k)$ query with a non-negligible advantage, then we can construct an algorithm B to solve the CDH problem with a non-negligible advantage.

Proof: The adversary sends a $\text{Test}(\prod_U^k)$ query to simulation B. If $\prod_U^k \neq \prod_{C'}^x$ and $\prod_U^k \neq \prod_{S'}^y$, simulation B aborts this session. Otherwise, we suppose that adversary can correctly guess the established session key in a non-negligible advantage. Let T denote the event that adversary sends a $\text{Test}(\prod_U^k)$ query. According to the formula of total probability, we can get

$$\begin{aligned} 1/2 + \varepsilon = {} & \Pr[A \text{ succeeds}] \\ = {} & \Pr[A \text{ succeeds} \mid C2S \cup T] \cdot \Pr[C2S \cup T] \\ & + \Pr[A \text{ succeeds} \mid \overline{C2S \cup T}] \cdot \text{pr}[\overline{C2S \cup T}] \end{aligned} \tag{8}$$

If adversary fails to impersonate client and fails to send an effective $\text{Test}(\prod_U^k)$ query, we can get

$$\Pr[A \text{ succeeds} \mid \overline{C2S \cup T}] = \tfrac{1}{2} . \tag{9}$$

Thus

$$\begin{aligned} \tfrac{1}{2} + \varepsilon \leq {} & \Pr[A \text{ succeeds} \mid C2S \cup T] \cdot \Pr[C2S \cup T] + \tfrac{1}{2} \\ \leq {} & \Pr[C2S \cup T] + \tfrac{1}{2} . \end{aligned} \tag{10}$$

Accordingly

$$\Pr[C2S] + \Pr[T] \geq \Pr[C2S \cup T] \geq \varepsilon .$$

Therefore

$$\Pr[T] \geq \varepsilon - \Pr[C2S] .$$

We have known that $\Pr[C2S]$ is negligible from Theorem 1. Therefore, the probability $\Pr[T]$ is non-negligible if adversary attacks successfully. Meanwhile, the probability of event $\prod_C^i = \prod_{C'}^x$ and $\prod_S^j = \prod_{S'}^y$ happen simultaneously is $1/q_n q_m$. The probability of achieving a successful session is $1/q_s^2$. As H_9 is a random oracle, simulation B can find K_{12}, R_{ui} and R'_{sj} in list L_9 in the probability of $1/q_9$. Therefore, we can compute xyP in the following equation:

$$\begin{gathered} xyP = K_{12} - aQ_2 - cQ_1 - acP \\ \text{or} \\ xyP = R'_{sj} - aQ_2 - d_{sj}Q_1 - ad_{sj}P \\ \text{or} \\ xyP = R_{ui} - cQ_1 - d_{ui}Q_2 - cd_{ui}P . \end{gathered} \tag{11}$$

Therefore, if adversary A can correctly guess the bit b in a non-negligible advantage ε, then simulation B can solve CDH problem in a non-negligible advantage

$$\frac{3\Pr[T]}{q_n q_m q_s^2 q_4} \geq \frac{3(\varepsilon - \Pr[C2S])}{q_n q_m q_s^2 q_4} \tag{12}$$

In summary, our protocol provides key agreement. □

Theorem 4: Our protocol provides implicit key confirmation.

Proof: Combining Theorems 1 and 2, our protocol can achieve mutual authentication under the CDH assumption in the modified BR93 model, while adversary cannot get session key according to Theorem 3. Therefore, we conclude that the proposed protocol offers implicit key confirmation. □

4.2.5 Full forward secrecy: The following theorem indicates that our protocol offers full forward secrecy.

Theorem 5: The proposed protocol provides full forward secrecy based on the CDH assumption.

Proof: In the proof of Theorem 3, adversary may send Corrupt queries on both $\prod_{C'}^x$ and $\prod_{S'}^y$, so that he can know the private keys of them. However, adversary is still unable to compute the associated established session keys of both $\prod_{C'}^x$ and $\prod_{S'}^y$. That is because *Test* query must be issued before *Corrupt* query in a session, thus Theorem 3 still holds under the *Corrupt* query on both $\prod_{C'}^x$ and $\prod_{S'}^y$. Hence, the corruption of both $\prod_{C'}^x$ and $\prod_{S'}^y$ does not help the adversary recovering the session keys. Therefore, we conclude that our protocol offers full forward secrecy. □

4.3 Security analysis

In this section, we illustrate that the proposed protocol can satisfy design goals described in Section 2.4.

© The Institution of Engineering and Technology 2019

17518717, 2020, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-ifs.2019.0249, Wiley Online Library on [01/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

4.3.1 Mutual authentication: According to Theorems 1 and 2, any probability polynomial adversary cannot masquerade as vehicle user or RSU to authenticate others. Specifically, both vehicle user and RSU verify each other's time key and certificate. We use notations A and B to represent two parties of communication. The party B firstly computes $R_B = (d_B + s_B) \cdot M_A$ and hashes it to get its certificate Auth_B, then B sends it to A. Next, party A verifies the certificate Auth_B by deriving Auth'_B from computing R'_B. Finally, A compares the value Auth'_B with its received message Auth_B.

During this process, due to the hardness of discrete logarithm (DL) problem, it is impossible to get $(r_A + s_A)$ and $(d_B + s_B)$ from $M_A = (r_A + s_A) \cdot P$ and $R_B = (d_B + s_B) \cdot M_A$, respectively. This guarantees the security of random numbers and private keys. Therefore, the protocol can provide mutual authentication.

4.3.2 Session key agreement: Through the above analysis of mutual authentication, both vehicle user and RSU can calculate the session key $\text{SK} = H_9(P_{\text{pub}}, P_{RA_{2k}}, P_{RA_{2e}}, \text{ID}_A, \text{PID}_B, M_A, M_B, R_A, R_B, K)$, which is used for future communication. Therefore, the proposed protocol can support session key agreement.

4.3.3 User anonymity: In the protocol, the real identity of vehicle user is masked by $\text{PID}_{ui} = ID_{ui} \oplus H_5(A_{ui})$, where $A_{ui} = (a + s_{ui}) \cdot P$. Therefore, recovering its real identity ID_{ui} needs the value A_{ui}. However, it is difficult to compute because $A_{ui} = (a + s_{ui}) \cdot P = (d_{\text{URA}_{2e}} + H_4(ID_{ui}))^{-1} \cdot B_{ui}$, in which random value a, s_{ui} and private key are not easily available due to the hardness of DL problem. Therefore, the proposed protocol can provide user anonymity.

4.3.4 Conditional privacy: On the prerequisite of protecting privacy for legitimate users, authority needs to recover the real identity of malicious user and take severe punishments to it. Specifically, the authority uses its private key d_{URA_2} to compute A_{ui}, and gets real identity by $\text{ID}_{ui} = \text{PID}_{ui} \oplus H_5(A_{ui})$. Therefore, the proposed protocol supports conditional privacy.

4.3.5 Flexibility of private key update: We enhance the flexibility of private key update through assigning different valid periods to vehicle user and RSU. Specially, registration authority selects time t_1 and t_2 for vehicle user and RSU, which embedded in their own time keys d_{tui} and d_{tsj}. Thus, entity can update its private key independently once time expires. Therefore, our protocol provides flexibility update for private keys.

Except for the above security goals, our protocol can also resist the well-known attacks.

4.3.6 Replay attack: Replay attack means that the valid data is maliciously repeated or delayed by adversary. When sponsoring a new request, the new random numbers $r_{ui}, r_{sj} \in Z_q^*$ are selected to get the element M_{ui} and M_{sj}. The value of M_{ui} and M_{sj} ensures the freshness of messages. Therefore, our protocol can resist replay attack.

4.3.7 Perfect forward security: Perfect forward security means that adversary cannot recover previously established session keys, even though private keys of entities have been corrupted. We use A and B to represent two parties of communication. In our protocol, A and B agree on a session key $\text{SK} = H_9(P_{\text{pub}}, P_{\text{TA}_{2k}}, P_{\text{TA}_{2e}}, \text{ID}_A, \text{PID}_B, M_A, M_B, R_A, R_B, K)$, where

$$\begin{aligned} M_A &= (r_A + s_A) \cdot P \\ R_A &= (d_A + s_A) \cdot M_B \\ K_A &= (r_A + s_A) \cdot M_B \end{aligned} \tag{13}$$

and M_B, R_B, K_B have a similar form. Assuming that entity A has been compromised, and its private key and random values r_A are obtained by adversary. To get session key SK, the adversary has to use s_A to compute M_B, R_B and K_B. However, it is difficult to derive s_A from $S_A = s_A \cdot P$ due to the hardness of DL problem. Therefore, our proposed protocol provides perfect forward security.

4.3.8 RALRI attack: There is no authority involved in the mutual authentication process, and only the vehicle user and RSU exchange parameters to ensure the fairness of our protocol. According to the analysis of perfect forward security, even though registration authority leaks the private key of entity, the security of session key can still be guaranteed. Therefore, our proposed protocol can resist the RALRI attack.

4.3.9 Known session key security: Due to the collision-resistant property of hash function, adversary fails to find some information from the session key $\text{SK} = H_9(\cdot)$. Therefore, our proposed protocol can achieve known session key security.

4.3.10 ESL attack: Combined secret values and random numbers, our protocol guarantees the security of private keys. Therefore, our proposed protocol can resist ESL attack.

5 Performance analysis

This section analyses the efficiency of our authentication protocol with other related protocols according to security, functionality and computation cost. It should be noted that only our protocol provides maximum facility to achieve secure communication.

To achieve a reliable security level of AES128 algorithm, we have utilised a Tate pairing on an MNT curve with embedding degree $k = 2$. Further, we have also chosen a $p = 1538$ bits finite field F_p and a group G_1 with order $q = 2^{255} + 2^{41} + 1$ that is generated from points on elliptic curve over a finite field F_p. We use the following cryptographic operations to measure and compare the computation cost.

- T_{pair}: Time for an admissible bilinear pairing operation $\hat{e}: G_1 \times G_1 \to G_2$.
- T_{mul}: Time for elliptic curve scalar point multiplication operation.
- T_{emul}: Time for point multiplication operation based on bilinear pairing.
- T_{mtp}: Time for a map-to-point hash function in G_1 of bilinear pairings setting.
- T_{exp}: Time for a modular exponentiation operation.
- T_{h}: Time for a general one-way hash function operation.

Cryptographic library MIRACL [22] is used to measure the execution time of the above-mentioned cryptographic operations. MIRACL is a C software library and regarded as the gold standard open source SDK for elliptic curve cryptography. The PC terminal and mobile device are adopted to simulate the computation power of RSU and OBU. The specific test environment is as follows:

- *PC terminal:* The test code runs on virtual machine Ubuntu 14.04.3 LTS. The physical machine is equipped with a Core 3G I7 2.40 GHz processor, 8G DDR3 memory and the Window 7 Ultimate, 64-bit 6.1.7601, Service Pack 1 operating system.
- *Mobile terminal:* The operating system is based on Android 6.0.1 MIUI 9 8.3.29 development version, the CPU is Qualcomm Snapdragon 801 (2.5 GHz) processor, 2G running memory.
- *Connection and compiler:* The connection tool is the Android Debug Bridge toolkit. The compiler environment for the PC and mobile devices is gcc version 4.8.4 and arm-none-linux-gnueabi cross-compilation tool, respectively.

© The Institution of Engineering and Technology 2019

17518717, 2020, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-ifs.2019.0249, Wiley Online Library on [01/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Table 1 Computation cost of cryptographic operations

Cryptographic operations	Presentation form	OBU time, s	RSU time, ms
T_{pair}	$\hat{e}(a \cdot P, b \cdot Q)$	0.181	63.4
T_{mul}	$x \cdot P$	0.032	13.4
T_{emul}	—	0.055	20.7
T_{mtp}	$H_1: \{0,1\}^* \rightarrow G_1$	0.273	114.2
T_{exp}	—	0.012	5.3
T_h	$H: \{0,1\}^* \rightarrow \{0,1\}^n$	0.001	0.13

Table 2 Functionality comparison

Functionality	P1	P2	P3	P4	P5	P6
[23]	✓	✓	×	×	×	×
[24]	×	✓	×	×	×	×
[25]	×	✓	×	✓	×	×
ours	✓	✓	✓	✓	✓	✓

P1: multi-authority structure, P2: mutual authentication, P3: conditional privacy, P4: independent update of private key, P5: resist RALRI attack, P6: resist ESL attack.

Table 3 Operation numbers in different protocols

Protocols	Operation numbers	
	OBU, s	RSU, ms
[23]	$2T_{\text{emul}} + 2T_{\text{mtp}} + T_{\text{pair}}(\simeq 0.84\,\text{s} = 840\,\text{ms})$	$2T_{\text{emul}} + 2T_{\text{mtp}} + T_{\text{pair}}(\simeq 333.2\,\text{ms})$
[24]	$T_{\text{mtp}} + 2T_{\text{pair}} + 5T_{\text{mul}}(\simeq 0.80\,\text{s} = 800\,\text{ms})$	$2T_{\text{pair}} + 4T_{\text{mul}}(\simeq 180.4\,\text{ms})$
[25]	$T_{\text{mtp}}(\simeq 0.27\,\text{s} = 270\,\text{ms})$	$T_{\text{pair}} + T_{\text{mul}} + T_{\text{exp}}(\simeq 82.1\,\text{ms})$
ours	$7T_{\text{mul}}(\simeq 0.22\,\text{s} = 220\,\text{ms})$	$7T_{\text{mul}}(\simeq 93.8\,\text{ms})$

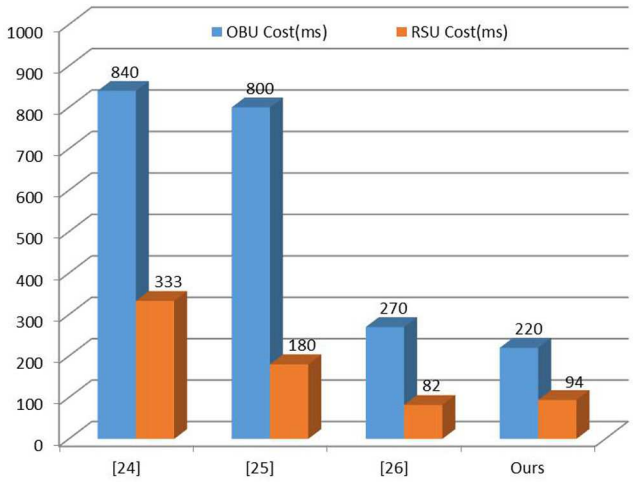

Fig. 6 *Computation cost of different schemes*

Detailed testing results about cryptographic operations are listed in Table 1.

The relatively lightweight operations such as general one-way hash function are ignored.

In Table 2, we compare the security requirements and functions of the proposed protocol with the existing related schemes [23–25]. For convenience, we let P1, P2, P3, P4, P5 and P6 represent these security requirements, respectively. In order to propose a secure and efficient scheme, the first goal is to satisfy the complete security requirements mentioned in Section 2, and the computation and communication cost may be increased due to desired property. In Table 2, all these cited schemes cannot resist cryptographic attacks such as ESL attack and RALRI attack, and cannot provide flexible update for private key. And only the scheme in [23] adopts the multi-authority structure. However, it is shown that all these security requirements are satisfied in our scheme.

In Table 3, the operation numbers of the proposed scheme with related schemes [23–25] are discussed. So, we further list the computation cost for OBU and RSU in Fig. 6. It is observed that our scheme is significantly superior to schemes [23, 24] for both RSU and OBU, while RSU in scheme [25] relatively spends less time than our scheme. However, all of these schemes are not suitable for practical implementation owing to the security loopholes listed in Table 2 except for our scheme. In short, our scheme has the best security with slightly compromise in efficiency.

6 Conclusion

To improve the security and efficiency of communication in VANET, we design a hierarchical revocable authentication protocol based on the SCPK and Schnorr signatures. In the proposed protocol, each registration authority can independently handle registration requests and thereby avoiding complete dependence on the single central authority. Meanwhile, it can resist the RALRI attack. The protocol can also resist the ESL attack by combining secret values with random numbers. Performances analysis shows that our protocol provides maximum facility according to computation cost and security property. It significantly reduces the

© The Institution of Engineering and Technology 2019

17518717, 2020, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-ifs.2019.0249, Wiley Online Library on [01/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

computation cost compared with most of the schemes. Therefore, our protocol is feasible and effective for the VANET environment.

7 Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (no. 2016YFB0800601), the Natural Science Foundation of China (no. 61303217,61502372), the Natural Science Foundation of Shaanxi province (no. 2013JQ8002,2014JQ8313) and the Guangxi Key Laboratory of Cryptography and Information Security (no. GCIS201802).

8 References

[1] Hartenstein, H., Laberteaux, L.P.: 'A tutorial survey on vehicular *ad hoc* networks', *IEEE Commun. Mag.*, 2008, **46**, (6), pp. 164–171
[2] Toh, C.K.: '*Ad Hoc mobile wireless networks: protocols and systems*' (Prentice Hall Ptr, Englewood Cliffs, NJ, USA, 2002)
[3] Martinez, F.J., Toh, C.K., Cano, J.C., *et al.*: 'Emergency services in future intelligent transportation systems based on vehicular communication networks', *IEEE Intell. Transp. Syst. Mag.*, 2010, **2**, (2), pp. 6–20
[4] 'Huwei'. Available at https://www.huawei.com/en/press-events/news/2018/6/Huawei-v2x-strategy-RSU-launch, accessed 12 March 2019
[5] 'WIRED'. Available at https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/, accessed 3 August 2019
[6] 'electrek'. Available at https://electrek.co/2019/04/01/tesla-autopilot-hacker-tricked/, accessed 3 August 2019
[7] Kenney, J.B.: 'Dedicated short-range communications (DSRC) standards in the United States', *Proc. IEEE*, 2011, **99**, (7), pp. 1162–1182
[8] Vijayakumara, P., Changb, V., Deboraha, L.J., *et al.*: 'Computationally efficient privacy preserving anonymous mutual and batch authentication schemes for vehicular Ad hoc networks', *Future Gener. Comput. Syst.*, 2016, **78**, pp. 943–955
[9] Farooq, S.M., Hussain, S.M., Kiran, S., *et al.*: 'Certificate based security mechanisms in vehicular ad-hoc networks based on IEC 61850 and IEEE WAVE standards', *Electronics*, 2019, **8**, (1), p. 96
[10] Tzeng, S., Horng, S., Li, T., *et al.*: 'Enhancing security and privacy for identity-based batch verification scheme in VANETs', *IEEE Trans. Veh. Technol.*, 2017, **66**, (4), pp. 3235–3248
[11] Fujioka, A., Suzuki, K.: 'Designing efficient authenticated key exchange resilient to leakage of ephemeral secret keys'. Int. Conf. on Topics in Cryptology: CT-RSA, San Fransisco, CA, USA, 2011
[12] Manvi, S.S., Tangade, S.: 'A survey on authentication schemes in VANETs for secured communication', *Veh. Commun.*, 2017, **9**, pp. 19–30
[13] Wasef, A., Shen, X.: 'EMAP: expedite message authentication protocol for vehicular *ad hoc* networks', *IEEE Trans. Mob. Comput.*, 2011, **12**, (1), pp. 78–89
[14] Chim, T.W., Yiu, S.M., Hui, L.C.K., *et al.*: 'Security and privacy issues for inter-vehicle communications in VANETs'. 2009 6th IEEE Annual Communications Society Conf. on Sensor, Mesh and Ad Hoc Communications and Networks Workshops, Rome, Italy, 2009, pp. 1–3
[15] Vighnesh, N.V., Kavita, N., Urs, S.R., *et al.*: 'A novel sender authentication scheme based on hash chain for vehicular ad-hoc networks'. 2011 IEEE Symp. on Wireless Technology and Applications (ISWTA), Langkawi, Malaysia, 2011, pp. 96–101
[16] Huang, J.L., Yeh, L.Y., Chien, H.Y., *et al.*: 'ABAKA: an anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks', *IEEE Trans. Veh. Technol.*, 2011, **60**, (1), pp. 248–262
[17] Sun, J., Zhang, C., Zhang, Y., *et al.*: 'An identity-based security system for user privacy in vehicular ad hoc networks', *IEEE Trans. Parallel Distrib. Syst.*, 2010, **21**, (9), pp. 1227–1239
[18] Lin, X., Sun, X., Ho, P.-H., *et al.*: 'GSIS: a secure and privacy-preserving protocol for vehicular communications', *IEEE Trans. Veh. Technol.*, 2007, **56**, (6), pp. 3442–3456
[19] Vijayakumar, P., Chang, V., Deborah, L.J., *et al.*: 'Computation ally efficient privacy preserving anonymous mutual and batch authentication schemes for vehicular ad hoc networks', *Future Gener. Comput. Syst.*, 2018, **78**, pp. 943–955
[20] Arain, Q.A., Zhongliang, D., Memon, I., *et al.*: 'Privacy preserving dynamic pseudonym-based multiple mix-zones authentication protocol over road networks', *Wirel. Pers. Commun.*, 2017, **95**, (2), pp. 505–521
[21] Bellare, M., Rogaway, P.: 'Entity authentication and key distribution'. Annual Int. Cryptology Conf., Santa Barbara, CA, USA, 1993, pp. 232–249
[22] 'MIRACL Cryptographic SDK'. Available at https://github.com/miracl/MIRACL, accessed 16 December 2018
[23] Tseng, Y.M., Huang, S.S., Tsai, T.T., *et al.*: 'List-free ID-based mutual authentication and key agreement protocol for multiserver architectures', *IEEE Trans. Emerg. Top. Comput.*, 2016, **4**, (1), pp. 102–112
[24] Han, M., Hua, L., Ma, S.: 'A self-authentication and deniable efficient group key agreement protocol for VANET', *KSII Trans. Internet Inf. Syst.*, 2017, **11**, (7), pp. 3678–3698
[25] Amin, R., Islam, S.H., Vijayakumar, P., *et al.*: 'A robust and efficient bilinear pairing based mutual authentication and session key verification over insecure communication', *Multimedia Tools Appl.*, 2018, **77**, (9), pp. 11041–11066

© The Institution of Engineering and Technology 2019

17518717, 2020, 1, Downloaded from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-ifs.2019.0249, Wiley Online Library on [01/07/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License