Hindawi
Mobile Information Systems
Volume 2019, Article ID 7016460, 9 pages
https://doi.org/10.1155/2019/7016460

Research Article

Authentication-Based Vehicle-to-Vehicle Secure Communication for VANETs

Huibin Xu,[1] **Mengjia Zeng**,[1,2] **Wenjun Hu**,[1] **and Juan Wang**[3]

[1]*School of Information Engineering, Huzhou University, Huzhou, Zhejiang Province 313000, China*
[2]*Qiuzhen School of Huzhou Teachers College, Huzhou, Zhejiang Province 313000, China*
[3]*School of Engineering, Huzhou University, Huzhou, Zhejiang Province 313000, China*

Correspondence should be addressed to Mengjia Zeng; zmj@zjhu.edu.cn

Received 8 April 2019; Revised 20 May 2019; Accepted 9 June 2019; Published 26 June 2019

Academic Editor: Alessandro Bazzi

Communication in VANETs is vulnerable to various types of security attacks since it is constructed based on an open wireless connection. Therefore, a lightweight authentication (LIAU) scheme for vehicle-to-vehicle communication is proposed in this paper. The LIAU scheme requires hash operations and uses cryptographic concepts to transfer messages between vehicles, in order to maintain the required security. Moreover, we made the LIAU scheme lightweight by introducing a small number of variable parameters in order to reduce the storage space. Performance analysis shows that the LIAU scheme is able to resist various types of security attacks and it performs well in terms of communication cost and operation time.

1. Introduction

Recently, vehicular ad hoc networks (VANETs) [1] have been favored by intelligent transportation system (ITS), and it is a part of ITS that aims to provide a safer, coordinated, and smarter mode of transportation. With the help of VANETs, ITS can improve traffic management efficiency and enhance road safety. VANETs use a vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication to obtain traffic and vehicle status information so that traffic accidents can be prevented and dealt in advance. Information among vehicles is exchanged through multihop transmission because V2V communication is based on dedicated short-range communication (DSRC) standard, which includes IEEE 802.11p. The vehicle is connected to the external network through the roadside units (RSUs). Figure 1 shows a typical structure of VANETs.

The primary purpose of deploying VANETs is to improve traffic safety. Transmitting messages efficiently and honestly among vehicles is the key to maintaining traffic safety [2]. However, VANET is in an open and insecure communication environment, which is vulnerable to various security attacks. For example, an attacker forges a road congestion message for his own benefit. The vehicle receiving the message mistakenly thinks that the road ahead is congested, and it makes a detour. So the attacker can seek personal gain.

Therefore, the vehicle needs to verify the received message and authenticate the sender. However, due to the mobility of the vehicles in the VANETs is usually very fast and the communication time among vehicles is short, vehicles need to be certified in a short time.

In addition, the vehicle may receive multiple messages at the same time. In a dense environment, a vehicle may simultaneously receive messages from a dozen or even dozens of other vehicles. Therefore, how to complete the authentication of multiple messages in a short time is an urgent problem.

To address this problem, a lightweight authentication (LIAU) scheme is proposed. The LIAU scheme introduces a simple two-layer model to authenticate V2V communication. It uses a hash function to generate system parameters. These parameters are used to authenticate the communication entities. Performance analysis shows that the proposed LIAU scheme can resist impersonation attack, modification attack,

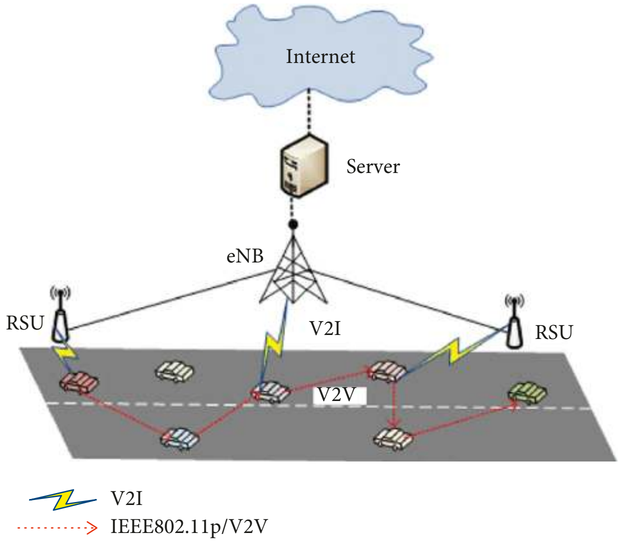


Figure 1: Typical structure of VANETs.

and replay attack. In addition, the LIAU scheme has low communication cost and operation time.

2. Related Work

It is easier for attackers to attack the VANETs because it transmits messages via a wireless medium. Once the network has been attacked, transmission delay will be long and the message may have been tampered, even lost [3]. A wrong and tampered message in VANETs may cause traffic congestion or even a traffic accident. Hence, the vehicle must carry on the authentication to the received message, and it should defend various attacks.

For the security of transmitted messages, Vijayakumar et al. [4] proposed a dual authentication and key management (DAKM) strategy. The DAKM strategy uses dual authentication to prevent unauthenticated vehicles from entering VANETs. In addition, the DAKM strategy effectively updates messages. However, it does not protect the location privacy of vehicles.

Chuang and Chen [5] proposed a trust-extended authentication (TEA) strategy to authenticate V2V communication entities. The TEA strategy uses historical trust relationships between vehicles to authenticate communication entities. However, it does not provide a specific way to authenticate messages. So, an internal attack may prevail. Therefore, Kumari et al. [6] proposed an enhanced TEA (E-TEA) strategy. Although E-TEA can defend against internal attacks, it has a long running time and has heavy computational burden.

Li et al. [7] proposed an authentication framework with conditional privacy preservation and nonrepudiation (ACPN). ACPN realizes the nonrepudiation of vehicles through the public key encryption-based pseudonym mechanism. Extendibility is an important feature of ACPN, and it is convenient for other systems. However, the storage cost of ACPN is high. Wang et al. [8] proposed a two-factor lightweight privacy-preserving authentication (TFLIP) strategy. It makes use of the two-factor biological encryption mechanism to authenticate the received messages. But, the security of the TFLIP strategy depends heavily on the system secret keys.

Lee and Lai [9] proposed a secure batch verification with the group testing (SBVGT) scheme to maintain the security of VANETs. However, the scheme can only defend against impersonation attack, but it cannot resist replay attack, and it is not traceable. In addition, Muthumeenakshi et al. [10] proposed a three-party password-based authenticated key exchanged (TPKE) strategy. However, the TPKE strategy does not analyze security attacks during the communication phase. Sun et al. [11] put forward privacy-preserving mutual authentication (PPMA) to resist a DoS attack. The PPMA strategy realizes conditional privacy verification by signature. But, it has high communication costs. Vasudev and Das [12] proposed a lightweight authentication protocol to protect V2V communication from various attacks. However, the authentication protocol has not specified what kind of encryption algorithm is used. In addition, it does not compare its security performance with similar authentication protocol. Ibrahim et al. [13] also emphasized the security of V2V communication and proposed central push-based replication protocol (CPRP) in order to improve the authentication service availability. But it strongly depends on RSUs, which increase the economic cost of deploying RSUs. Malik and Pandey [14] proposed a threat driven authentication approach based on discrete event. It used Petri nets to implement the authentication, which increased communication overhead.

3. System Model

3.1. Network Model. Consider a simple two-layer network model, as shown in Figure 2. Regional authorities (RAs) lie on the top layer, and vehicles are on the ground layer. Assume that RA is the fully trusted manager, and they are distributed authorities. Each authority covers a region. And RA is in charge of generating system parameters, and it is responsible for registering vehicles. The registered vehicles are allowed to enter the network.

All vehicles in the system are equipped with the tamper-proof device (TPD), which is used to store encrypted data, including secret key and pseudonym. However, the parameter of TPD is assigned by RA. At the same time, assume that TPD has the highest security level, which can defend against any attack. In addition, each vehicle is fitted with on-board unit (OBU). Vehicles transmit and receive messages with the help of OBU. In addition, Table 1 shows the main notations and their corresponding meanings.

3.2. Attack Model. Assume that RA has the highest security level and it can defend against any attack. This paper only considers two types of attacks [15]:

(1) External attack: this attack refers to that the unregistered vehicle (external) attacks the network system by various means. Such as replay attack, tracking attack, and impersonation attack.

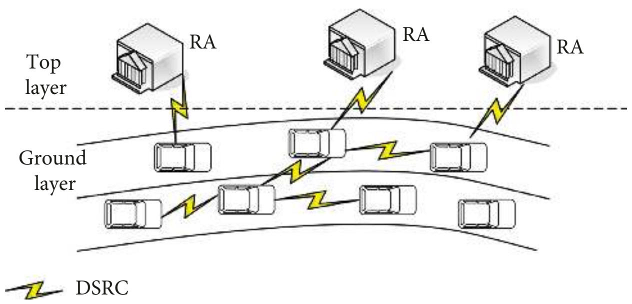


Figure 2: Network model.

Table 1: The notations and specific descriptions.

Notations	Description
OBU	On board unit
RSU	Roadside unit
RA	Regional authority
ϑ_x	A vehicle x in the network
$\mathrm{ID}_{\vartheta_x}$	ID of ϑ_x
$\mathrm{ID}_{\mathrm{RA}}$	ID of RA
$h(\cdot)$	Hash function
κ_x	Private key of eneity x
$\|$	The connection symbol
$\oplus$	The XOR operator

(2) Internal attack: internal attack refers to seeking personal gains by releasing false information and disguising the identity of registered vehicles. Internal attacks result from a small number of maliciously registered vehicles.

4. LIAU Scheme

The LIAU scheme aims to achieve lightweight certification of V2V communications so that the communication vehicles are legitimate. In other words, only certificated vehicles are allowed to communicate with other vehicles. The LIAU scheme consists of the initial stage, the registration stage, and the authentication stage.

4.1. Initial Stage. In the LIAU scheme, each RA has a unique identity (ID). RA generates its own private key using a secure single hash function $h(\cdot)$:

$$\kappa_{\mathrm{RA}} = h\left(\mathrm{ID}_{\mathrm{RA}} \| S_{\mathrm{RA}}\right), \tag{1}$$

where κ_{RA} is the private key to RA. And $\mathrm{ID}_{\mathrm{RA}}$ represent the ID of RA. S_{RA} is the random number generated by RA.

A MD5 algorithm is one of the most common hash functions [16]. It takes as input a message of arbitrary length and generates as output a 128 bit message digest. The basic principle of the MD5 algorithm is to divide the input message into blocks with 512 bits, and each block is divided into 16 subblocks with 32 bits. After a series of processing, the output consists of four groups with 32 bit. The four groups are cascaded, and a hash value with 128 bits is generated.

MD5, SHA-1, and SHA-2 are one-way hash functions. Compared with SHA-1 and SHA-2, the efficiency of MD5 is more. Under the same conditions, the execution time of MD5 is 0.000007 s [17]. However, the execution time of SHA-1 and SHA-2 algorithms is up to 0.00018 s and 10.150778 s [17], respectively. This is why the MD5 algorithm was chosen for the LIAU scheme.

4.2. Registration Stage. Similarly, each vehicle in the LIAU scheme has a unique ID and secret key. Let $\mathrm{ID}_{\vartheta_a}$ represent the ID of the vehicle ϑ_a. Let κ_{ϑ_a} represent the secret key of the vehicle ϑ_a. Before the vehicle is registered with the system, it will compute the parameters using $\mathrm{ID}_{\vartheta_a}$ and κ_{ϑ_a}, as shown in the following equation:

$$P_{\vartheta_a} = h\left(\mathrm{ID}_{\vartheta_a} \| \kappa_{\vartheta_a}\right). \tag{2}$$

Then, the vehicle ϑ_a computes the parameter ξ_{ϑ_a}, as shown in the following equation:

$$\xi_{\vartheta_a} = A_{\vartheta_a} \oplus B_{\vartheta_a}, \tag{3}$$

where the symbol "$\oplus$" represents XOR operation. And $A_{\vartheta_a} = h(\mathrm{ID}_{\vartheta_a} \| P_{\vartheta_a})$ and $B_{\vartheta_a} = h(\kappa_{\vartheta_a} \| P_{\vartheta_a})$. Finally, the parameter ξ_{ϑ_a} is transmitted toward RA by using the vehicle ϑ_a.

Once received, the RA first generates a random number $\hbar_{\mathrm{RA}}$. Then, the RA computes the parameter π_{RA}:

$$\pi_{\mathrm{RA}} = h\left(\xi_{\vartheta_a} \| \psi_{\mathrm{RA}}\right) \oplus \kappa_{\mathrm{RA}}, \tag{4}$$

where $\psi_{\mathrm{RA}} = h(\mathrm{ID}_{\mathrm{RA}} \| \hbar_{\mathrm{RA}})$. Finally, the parameters π_{RA} and $\hbar_{\mathrm{RA}}$ are transmitted toward the vehicle ϑ_a. Once received, the vehicle ϑ_a stores these parameters in TPD. And the vehicle ϑ_a forms a parameters set $\left\{\xi_{\vartheta_a}, \pi_{\mathrm{RA}}, \hbar_{\vartheta_a}, \hbar_{\mathrm{RA}}\right\}$. The entire registration process is shown in Figure 3, when the vehicle gets its own registered parameters $\left\{\xi_{\vartheta_a}, \pi_{\mathrm{RA}}, \hbar_{\vartheta_a}, \hbar_{\mathrm{RA}}\right\}$, it should make these parameters stored in TPD.

4.3. Authentication Stage

4.3.1. Identity-Oriented Initial Detection. Before communicating with other vehicles, the vehicle first authenticates its identity by itself and can only communicate with other vehicles after completing the authentication stage [12]. The vehicle generates the parameter ξ using its own ID and its secret keys, as shown in equations (2) and (3). The generated parameter ξ will be compared with the parameter stored in TPD. If they are the same, the vehicle succeeds to authenticate. If they are not consistent, the vehicle has to reregister with VS until the authentication succeeds.

Specifically, if the vehicle ϑ_a needs to communicate with other entities, it recalculates the parameter ξ'_{ϑ_a} according to equation (3). Specifically speaking, the vehicle ϑ_a computes $P'_{\vartheta_a} = h(\mathrm{ID}_{\vartheta_a} \| \kappa_{\vartheta_a})$, $A'_{\vartheta_a} = h(\mathrm{ID}_{\vartheta_a} \| P'_{\vartheta_a})$, and $B'_{\vartheta_a} = h(\kappa_{\vartheta_a} \| P'_{\vartheta_a})$. Then, the recalculated parameter $\xi'_{\vartheta_a} = A'_{\vartheta_a} \oplus B'_{\vartheta_a}$ is compared with parameter ξ_{ϑ_a} stored in TPD. If they are same, the vehicle ϑ_a is authenticated successfully, and it is allowed to communicate with other entities.

It is worth noting that the vehicle certification process is relatively simple, and each vehicle only needs to verify the

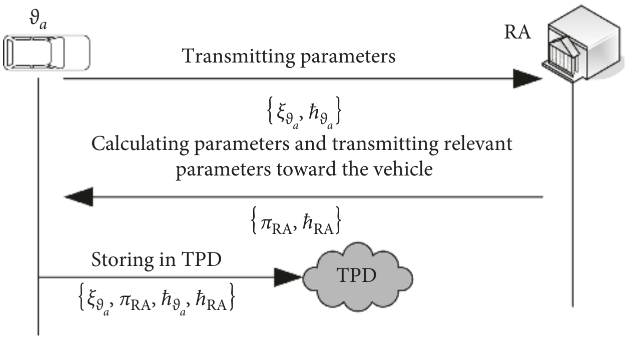

ϑ_a
RA
Transmitting parameters
$\{\xi_{\vartheta_a}, \hbar_{\vartheta_a}\}$
Calculating parameters and transmitting relevant parameters toward the vehicle
$\{\pi_{\mathrm{RA}}, \hbar_{\mathrm{RA}}\}$
Storing in TPD
$\{\xi_{\vartheta_a}, \pi_{\mathrm{RA}}, \hbar_{\vartheta_a}, \hbar_{\mathrm{RA}}\}$
TPD

Figure 3: Registration stage of vehicles.

parameters generated again with the parameters stored in TPD. If so, the vehicle is considered to be registered and authenticated successfully. However, the vehicles that are authenticated does not mean that they are a nonattacker. In fact, the authentication process would only make sense for nonattacking vehicles. These nonattacking vehicles are authenticated to ensure that the parameters acquired during the registration phase are correct.

4.3.2. Control Message-Based Authentication. In order to ensure the security of transmitting data, the communication entity needs to be verified before it is ready to transmit data.

(1) Request Message. Specifically, when the vehicle ϑ_a needs to transmit data to the vehicle ϑ_b, it first sends a request message (Rqst) to the vehicle and records the timestamp sent Rqst. At the same time, the vehicle ϑ_a generates a random number $\hbar^1_{\vartheta_a}$. Then, the vehicle extracts the parameters from TPD, and the value of the parameter ψ_{RA} is calculated as $\psi_{\mathrm{RA}} = h(\mathrm{ID}_{\mathrm{RA}} \| \hbar_{\mathrm{RA}})$.

The vehicle ϑ_a makes use of the parameters ξ_{ϑ_a}, ψ_{RA}, and π_{RA}, to compute the secret key of RA, as shown in the following equation:

$$\kappa_{\mathrm{RA}} = h\left(\xi_{\vartheta_a} \| \psi_{\mathrm{RA}}\right) \oplus \pi_{\mathrm{RA}}. \tag{5}$$

After that, the vehicle ϑ_a calculates the following parameters:

$$\Gamma_{\vartheta_a} = h\left(\kappa_{\mathrm{RA}} \| T_{\mathrm{te}}\right) \oplus \hbar^1_{\vartheta_a}, \tag{6}$$

$$\mathrm{M}_{\vartheta_a} = \Gamma_{\vartheta_a} \oplus \hbar^1_{\vartheta_a} \oplus \kappa_{\mathrm{RA}}, \tag{7}$$

$$\Upsilon_{\vartheta_a} = \mathrm{Rqst} \oplus \Gamma_{\vartheta_a} \oplus \kappa_{\mathrm{RA}} \oplus T_{\mathrm{te}}. \tag{8}$$

Finally, the vehicle ϑ_a transmits parameters $\{\Gamma_{\vartheta_a}, \Upsilon_{\vartheta_a}, T_{\mathrm{te}}\}$ toward the vehicle ϑ_b, where T_{te} is the timestamp that transmitted Rqst.

(2) Reply Message. The vehicle ϑ_b first records the timestamp of the received parameters $\{\Gamma_{\vartheta_a}, \Upsilon_{\vartheta_a}, T_{\mathrm{te}}\}$, which is marked as T_{re}. Then, T_{re} is compared with T_{te} that was extracted from $\{\Gamma_{\vartheta_a}, \Upsilon_{\vartheta_a}, T_{\mathrm{te}}\}$.

If T_{re} is too late, then the following inequality should hold:

$$T_{\mathrm{re}} - T_{\mathrm{te}} \geq \Delta T_1, \tag{9}$$

where ΔT_1 is the system parameter. When inequality (9) holds, it means that the received parameters $\{\Gamma_{\vartheta_a}, \Upsilon_{\vartheta_a}, T_{\mathrm{te}}\}$ have expired. And the vehicle ϑ_b immediately stops communicating with vehicle ϑ_a. Otherwise, go to the next step. The vehicle ϑ_b recalculated the parameter $\hbar^1_{\vartheta_a}$, which is already generated by the vehicle ϑ_a. The recalculated parameter $\hbar^1_{\vartheta_a}$ is given by

$$\widehat{\hbar}^1_{\vartheta_a} = \Gamma_{\vartheta_a} \oplus h\left(\kappa_{\mathrm{RA}} \| T_{\mathrm{te}}\right). \tag{10}$$

Similarly, the vehicle ϑ_b recalculates $\widehat{\mathrm{M}}_{\vartheta_a}$, as shown in the following equality:

$$\widehat{M}_{\vartheta_a} = \widehat{\hbar}^1_{\vartheta_a} \oplus \Gamma_{\vartheta_a} \oplus \kappa_{\mathrm{RA}}. \tag{11}$$

Then, the vehicle ϑ_b extracts Rqst message from $\Upsilon_{\vartheta_a} = \mathrm{Rqst} \oplus \Gamma_{\vartheta_a} \oplus \kappa_{\mathrm{RA}} \oplus T_{\mathrm{te}}$, which is given by

$$\mathrm{Rqst} = \Upsilon_{\vartheta_a} \oplus \Gamma_{\vartheta_a} \oplus \kappa_{\mathrm{RA}} \oplus T_{\mathrm{te}}. \tag{12}$$

After obtaining these parameters, the vehicle ϑ_b calculates two new parameters H_{ϑ_b} and I_{ϑ_b}, which are given by

$$H_{\vartheta_b} = h\left(\widehat{\hbar}^1_{\vartheta_a} \| \Delta T_1 \| \kappa_{\mathrm{RA}}\right), \tag{13}$$

$$I_{\vartheta_b} = H_{\vartheta_b} \oplus \kappa_{\mathrm{RA}} \oplus \widehat{M}_{\vartheta_a} \oplus \widehat{\hbar}^1_{\vartheta_a}. \tag{14}$$

Finally, the vehicle ϑ_b transmits the relevant parameters toward the vehicle ϑ_a. Once received, the vehicle ϑ_a sends a reply message to the vehicle ϑ_b. Considering the security of the channel, the reply message is encrypted, which is given by

$$\mathrm{EN\_Reply} = \mathrm{ENC}_{H_{\vartheta_b}}(\mathrm{Reply}). \tag{15}$$

At last, the vehicle ϑ_b transmits $\{I_{\vartheta_b}, T_{\mathrm{re}}, \mathrm{Reply}\}$ toward the vehicle.

The purpose of this paper is to reduce the operation time of the authentication scheme and make the scheme "lightweight." The lightweight RC4 algorithm meets the requirements. In essence, RC4 is a variable key-size stream cipher algorithm with high efficiency and good nonlinearity [18]. Compared with similar symmetric encryption algorithms A5 and CRC32, the RC4 algorithm has shorter operation time.

The three phases of RC4 operation are the state initialization, key-scheduling algorithm (KSA), Algoirthm 1, and pseudo random generation algorithm (PGRA), Algoritthm 2. The execution process of RC4 is shown in Figure 4. KSA generates initial 256 bytes permutation state, which is the input of PGRA. And the keystream is generated by using the PRGA.

Recall equation (15); the data that need to be encrypted are Reply messages, and the key is H_{ϑ_b}. Using Reply and H_{ϑ_b} as inputs to RC4, the encrypted EN_Reply can be generated.

(3) Authentication of Communication Entities. By exchanging control packets (Rqst, Reply) between ϑ_a and ϑ_b, they obtain each other's information. Once receiving

```
unsigned char s[256]
   char key[256]
len = strlen (key)
      void RC4_init (unsigned char * s, unsigned char * key)
{int i = 0, int j = 0
unsigned char k[256] = {0};
unsigned char tmp = 0;
   for (i = 0; i < 256; i++) {s[i] = i; k[i] = key[i%len];}
   for (i = 0; i < 256; i++) {j = (j + s[i] + k[i])%256;
                                tmp = s[i]; s[i] = s[j]; s[j] = tmp;}
}
```

Algorithm 1: KSA.

```
unsigned char s[256]
      void RC4_PGRA (unsigned char * s, unsigned char * data)
{int i = 0, int j = 0, int t = 0;
unsigned long k = 0;
unsigned char tmp = 0;
   len = strlen (data)
for (k = 0; k < len; k++) {i = (i + 1)%256; j = (j + s[i])%256;
                            tmp = s[i]; s[i] = s[j];
                            s[j] = tmp;
                            t = (s[i] + s[j])%256;
                            Data[k]^= s[t];}
}
```

Algorithm 2: PGRA.

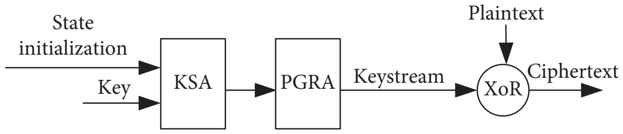


Figure 4: The execution process of the RC4 algorithm.

$\{I_{\vartheta_b}, T_{\text{re}}, \text{EN\_Reply}\}$, the vehicle ϑ_a first records the time of receiving $\{I_{\vartheta_b}, T_{\text{re}}, \text{EN\_Reply}\}$. And the timestamp is denoted as T_{re}^a. Then, the vehicle ϑ_a checks whether inequality $T_{\text{re}}^a - T_{\text{re}} \geq \Delta T_2$ is satisfied. If not, the vehicle ϑ_a stops communicating with the vehicle ϑ_b.

When inequality $T_{\text{re}}^a - T_{\text{re}} \geq \Delta T_2$ holds, the vehicle ϑ_a will extract Reply from EN_Reply. To extract Reply, the vehicle ϑ_a must calculate H_{ϑ_b} correctly and decrypt successfully EN_Reply. Accordingly, the vehicle ϑ_a calculates H_{ϑ_b} according to equation (16). Let $\widehat{H}_{\vartheta_b}$ represent H_{ϑ_b} calculated by the vehicle ϑ_a, which is given by

$$\widehat{H}_{\vartheta_b} = I_{\vartheta_b} \oplus \kappa_{RA} \oplus M_{\vartheta_a} \oplus \hbar_{\vartheta_a}. \tag{16}$$

Then, the parameter $\widehat{H}_{\vartheta_b}$ is used to decrypt EN_Reply and extract successfully Reply, which is given by

$$\text{Reply} = \text{DEC}_{\widehat{H}_{\vartheta_b}}(\text{EN\_Reply}), \tag{17}$$

where $\text{DEC}_{\widehat{H}_{\vartheta_b}}(\cdot)$ is the decrypted function. If $\widehat{H}_{\vartheta_b} == H_{\vartheta_b}$ holds, the vehicle ϑ_a can decrypt EN_Reply and extract Reply. Once properly decrypted, the vehicle ϑ_a considers that the vehicle ϑ_b is secure. And the vehicle ϑ_a would communicate with the vehicle ϑ_b. The entire process is shown in Figure 5.

5. Security Analysis

The formal expression of a security analysis model [19] is used to discuss the security of the LIAU system, aiming to verify that LIAU can resist common security attacks in VANETs.

5.1. Impersonation Attack.

5.1. Impersonation Attack. If an attacker $\wp$ is interested in other user's dedicated service, the attacker $\wp$ can impersonate the identity of another user and forge a valid login request. If an attacker $\wp$ can successfully forge, it may have successfully launched an impersonation attack.

In the LIAU scheme, in order to send a valid request, an attacker $\wp$ must forge an unassailable request message Rqst. Accordingly, the attacker $\wp$ needs to steal the parameters $\{\kappa_{\text{RA}}, \Upsilon_{\vartheta_a}, M_{\vartheta_a}, T_{\text{te}}\}$. However, it is very difficult for the attacker $\wp$ to steal these parameters. Even if, in some cases, the attacker $\wp$ has obtained the secret key κ_{RA} of RA, the attacker $\wp$ cannot steal the parameters $\{\Upsilon_{\vartheta_a}, M_{\vartheta_a}\}$.

According to equation (8), if the attacker $\wp$ wants to calculate Υ_{ϑ_a}, the attacker must know M_{ϑ_a}. To calculate it, the

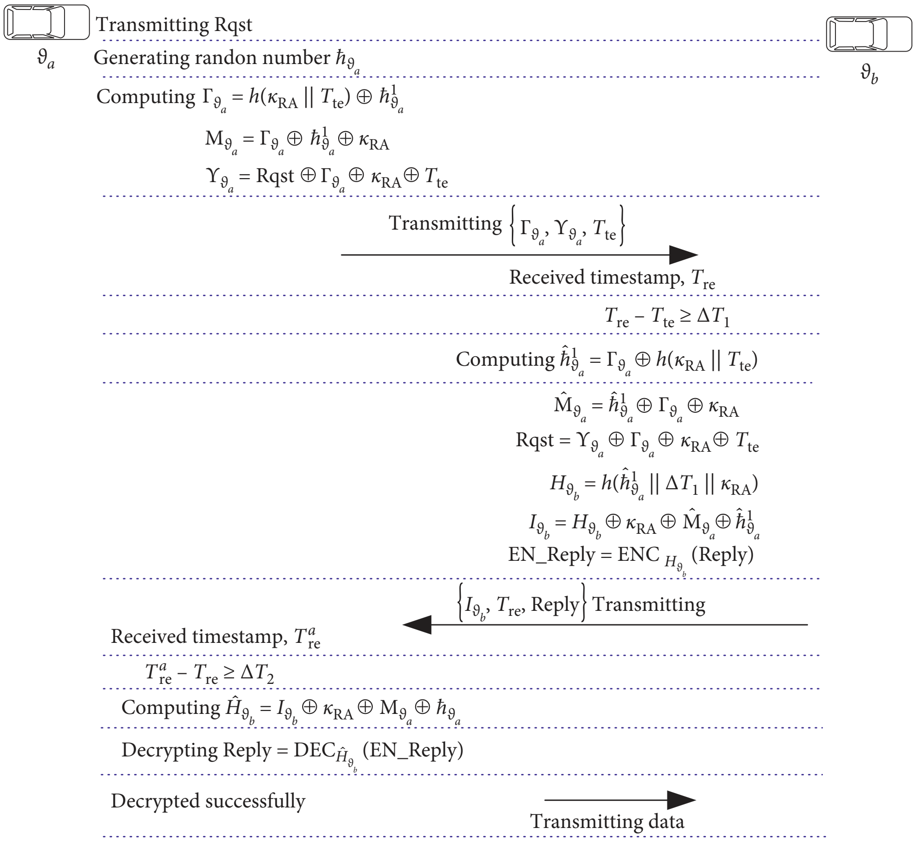

Transmitting Rqst
ϑ_a
ϑ_b
Generating randon number $\hbar_{\vartheta_a}$
Computing $\Gamma_{\vartheta_a} = h(\kappa_{\mathrm{RA}} \,||\, T_{\mathrm{te}}) \oplus \hbar^1_{\vartheta_a}$
$\mathrm{M}_{\vartheta_a} = \Gamma_{\vartheta_a} \oplus \hbar^1_{\vartheta_a} \oplus \kappa_{\mathrm{RA}}$
$\Upsilon_{\vartheta_a} = \mathrm{Rqst} \oplus \Gamma_{\vartheta_a} \oplus \kappa_{\mathrm{RA}} \oplus T_{\mathrm{te}}$
Transmitting $\left\{\Gamma_{\vartheta_a}, \Upsilon_{\vartheta_a}, T_{\mathrm{te}}\right\}$
Received timestamp, T_{re}
$T_{\mathrm{re}} - T_{\mathrm{te}} \geq \Delta T_1$
Computing $\hat{\hbar}^1_{\vartheta_a} = \Gamma_{\vartheta_a} \oplus h(\kappa_{\mathrm{RA}} \,||\, T_{\mathrm{te}})$
$\hat{\mathrm{M}}_{\vartheta_a} = \hat{\hbar}^1_{\vartheta_a} \oplus \Gamma_{\vartheta_a} \oplus \kappa_{\mathrm{RA}}$
$\mathrm{Rqst} = \Upsilon_{\vartheta_a} \oplus \Gamma_{\vartheta_a} \oplus \kappa_{\mathrm{RA}} \oplus T_{\mathrm{te}}$
$H_{\vartheta_b} = h(\hat{\hbar}^1_{\vartheta_a} \,||\, \Delta T_1 \,||\, \kappa_{\mathrm{RA}})$
$I_{\vartheta_b} = H_{\vartheta_b} \oplus \kappa_{\mathrm{RA}} \oplus \hat{\mathrm{M}}_{\vartheta_a} \oplus \hat{\hbar}^1_{\vartheta_a}$
$\mathrm{EN\_Reply} = \mathrm{ENC}_{H_{\vartheta_b}}(\mathrm{Reply})$
$\left\{I_{\vartheta_b}, T_{\mathrm{re}}, \mathrm{Reply}\right\}$ Transmitting
Received timestamp, T^a_{re}
$T^a_{\mathrm{re}} - T_{\mathrm{re}} \geq \Delta T_2$
Computing $\hat{H}_{\vartheta_b} = I_{\vartheta_b} \oplus \kappa_{\mathrm{RA}} \oplus \mathrm{M}_{\vartheta_a} \oplus \hbar_{\vartheta_a}$
Decrypting Reply = $\mathrm{DEC}_{\hat{H}_{\vartheta_b}}(\mathrm{EN\_Reply})$
Decrypted successfully
Transmitting data

Figure 5: Schematic diagram of transmitting data.

attacker needs to get the parameters. But it is generated by the random parameters. Therefore, it is difficult for an attacker to forge a valid request message.

5.2. Replay Attack. The replay attack refers to attacking the system by resending others' information packets [20]. When an attacker $\wp$ obtains the information data that are transmitted to the vehicle ϑ_b from the vehicle ϑ_a, then the attacker will transmit the obtained information to the vehicle ϑ_b. In this case, the data is mistakenly sent to the attacker $\wp$, which is originally transmitted from the vehicle ϑ_a. The attacker then successfully performs the replay attack.

An attacker $\wp$ would launch a replay attack to delay or even stop the response to any request message. If a vehicle receives the request sent by the attacker $\wp$, it means that the attacker $\wp$ successfully launched the replay attack.

According to the message transmission strategy in Figure 5, the vehicle ϑ_a does not directly transmit the request message Rqst but indirectly makes Rqst embedding in the parameter Υ_{ϑ_a}. And the transmitted message carries a timestamp. Once receiving Rqst, the vehicle ϑ_b first judges whether $T_{\mathrm{re}} - T_{\mathrm{te}} \geq \Delta T_1$ is satisfied. When not satisfied, the vehicle ϑ_b stops communicating. Therefore, it is difficult for an attacker $\wp$ to delay the request message.

Even if the attacker $\wp$ has received the message that is transmitted by the vehicle ϑ_b from the vehicle ϑ_a, and it has obtained $\mathrm{EN\_Reply} = \mathrm{ENC}_{H_{\vartheta_b}}(\mathrm{Reply})$. However, the attacker $\wp$ can only extract data from EN_Reply if it computes H_{ϑ_b} correctly. According to equation (13), it can be known that the attacker $\wp$ can only compute H_{ϑ_b} correctly if the attacker $\wp$ has known for the relevant parameters of the vehicle ϑ_a, namely, $\left\{I_{\vartheta_b}, \kappa_{\mathrm{RA}}, M_{\vartheta_a}, \hbar_{\vartheta_a}\right\}$. But, the attacker $\wp$ does not get these parameters simultaneously. So, it is difficult for an attacker $\wp$ to launch a replay attack on the system.

5.3. Tampering Attack. Tampering attack refers to that an attacker $\wp$ tampers other users' communication data. For an attacker $\wp$, it may launch a tampering attack if it can change data illegally.

Taking communication between vehicle ϑ_a and vehicle ϑ_b as an example, the defense tampering attack performance of the LIAU scheme is analyzed. Suppose the attacker $\wp$ has tampered the parameters $\left\{\Gamma_{\vartheta_a}, \Upsilon_{\vartheta_a}, T_{\mathrm{te}}\right\}$, which are transmitted by the vehicle ϑ_a from the vehicle ϑ_b. This delivers inaccurate data to the vehicle ϑ_b. The vehicle ϑ_b still calculates the relevant parameters using the mistaken parameters, including H'_{ϑ_b} and I'_{ϑ_b}, because the vehicle ϑ_b is not aware of the error. In addition, the vehicle ϑ_b encrypts these parameters using H'_{ϑ_b}. And these parameters are transmitted toward the vehicle ϑ_a.

As shown in Figure 6, the vehicle ϑ_a still recalculated H_b using its original parameters. Then, it checks whether $H'_{\vartheta_b} \neq H_{\vartheta_b}$ is satisfied. If satisfied, the vehicle ϑ_a does not transmit any data to the vehicle ϑ_b. H'_{ϑ_b} is surely different with H_{ϑ_b} because the parameters $\left\{\Gamma_{\vartheta_a}, \Upsilon_{\vartheta_a}, T_{\mathrm{te}}\right\}$ are changed

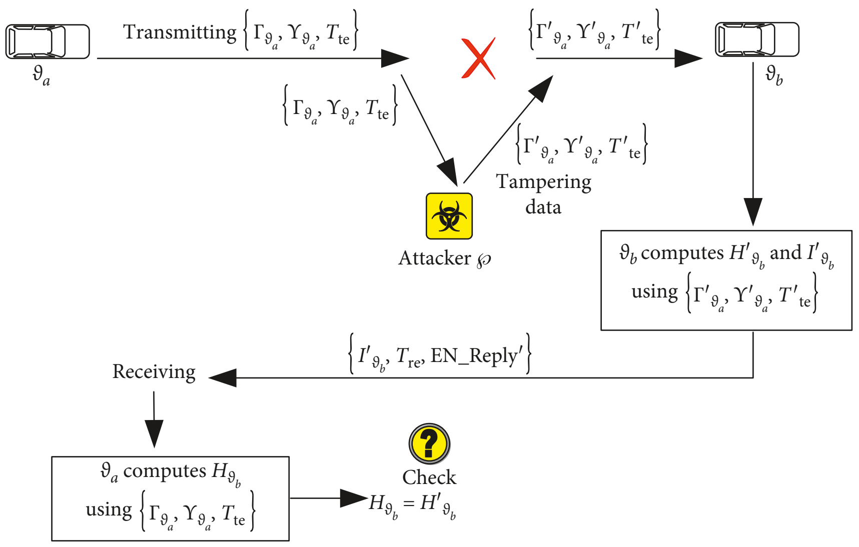

Transmitting $\{\Gamma_{\vartheta_a}, \Upsilon_{\vartheta_a}, T_{\mathrm{te}}\}$
ϑ_a
$\{\Gamma'_{\vartheta_a}, \Upsilon'_{\vartheta_a}, T'_{\mathrm{te}}\}$
ϑ_b
$\{\Gamma_{\vartheta_a}, \Upsilon_{\vartheta_a}, T_{\mathrm{te}}\}$
$\{\Gamma'_{\vartheta_a}, \Upsilon'_{\vartheta_a}, T'_{\mathrm{te}}\}$
Tampering data
Attacker $\wp$
ϑ_b computes H'_{ϑ_b} and I'_{ϑ_b} using $\{\Gamma'_{\vartheta_a}, \Upsilon'_{\vartheta_a}, T'_{\mathrm{te}}\}$
$\{I'_{\vartheta_b}, T_{\mathrm{re}}, \mathrm{EN\_Reply}'\}$
Receiving
ϑ_a computes H_{ϑ_b} using $\{\Gamma_{\vartheta_a}, \Upsilon_{\vartheta_a}, T_{\mathrm{te}}\}$
Check
$H_{\vartheta_b} = H'_{\vartheta_b}$

FIGURE 6: Defense tampering attack of the LIAU scheme.

by an attacker $\wp$. Therefore, the LIAU scheme is able to defend against impersonation attacks.

5.4. Comparison of Security Performance. Table 2 lists the performance of the representative authentication scheme mentioned in related work. The performance of defending against impersonation attack, replay attack, and tampering attack is analyzed. These three types of attacks are common in VANETs, and most strategies are resistant to them. Unsurprisingly, the proposed LIAU strategy also has the ability to resist these attacks. This also shows that the LIAU strategy meets the basic security of VANETs.

6. Performance Analysis

This section discusses the communication cost, storage cost, and operation time of the LIAU scheme. The computer parameters used for this performance analysis are as follows: Intel (R) Core (TM) i5-7500 CPU, 3.40 GHz, and RAM 8.00 GB.

6.1. Communication and Storage Costs. Firstly, the communication cost and storage cost of the LIAU scheme are analyzed. The communication cost refers to the communication overhead that results from computing and exchanging the parameters during the V2V communication stage. And the storage cost is the amount of space required to store all parameters. In addition, assume that the hash digest size of SHA-2 is 32 bytes, size of the ID number and the random number are 8 bytes, and size of the timestamp is 4 bytes. Size of bilinear pairings is 128 bytes. Operation of symmetric and asymmetric encryption or decryption requires 64 bytes. Signature operation requires 128 bytes.

Figure 7 shows the communication and storage costs for the LIAU scheme. As can be seen from Figure 7, communication cost and storage cost of the LIAU scheme remain the lowest, compared with ACPN, TFLIP, E-TEA, TPKE, and PPMA. This is in line with the original intention of designing the LIAU strategy, which reduces communication and storage costs.

6.2. Operation Time. Operation time refers to the time that the vehicle has taken to register, authenticate, and communicate. The longer the operation time, the more complex the algorithm is. Different schemes implement different operations. Let $T_{\mathrm{h}}(\cdot)$ represent the time taken to perform a one-way hash operation. Let T_{asen} and T_{aden} be the time taken to execute symmetric encryption and decryption, respectively. Let T_{s} represent the time taken to execute the signature operation. In addition, T_{e} and T_{b} are the time taken to execute the exponential operation and bilinear pairing, respectively. These parameter values are as follows: $T_{\mathrm{h}} = 0.0004$ ms, $T_{\mathrm{asen}} = 0.0800$ ms, $T_{\mathrm{aden}} = 1.46$ ms, $T_{\mathrm{s}} = 1.48$ ms, $T_{\mathrm{e}} = 0.600$ ms, and $T_{\mathrm{b}} = 1.600$ ms.

Figure 8 shows the operation time of the LIAU scheme. As can be seen from Figure 8, the operation time of the LIAU scheme is relatively short, compared with that of TFLIP, E-TEA, TPKE, and PPMA schemes. Although the operation time of the ACPN scheme is lower than that of other schemes, its communication cost and storage cost are high. In other words, the operation time of the ACPN scheme is lower at the cost of high communication and storage cost.

7. Conclusion and Future Work

The intermittent nature of V2V communications poses a challenge to authenticate the communication entity and exchanged messages among vehicles. Therefore, a lightweight authentication scheme for V2V communication (LIAU) is proposed. The LIAU scheme only has used the hash operation to maintain the security of the message transmission. And it has introduced a small number of variable parameters in order to reduce the storage space and operation time. Performance analysis shows that the LIAU scheme can resist common security attacks in VANETs.

Table 2: Comparison of security performance.

Performance	SBVGT	ACPN	TELIP	E-TEA	TPKE	PPMA	LIAU
Defending against impersonation attack	✓	✓	✓	✓	✓	✓	✓
Defending against tampering attack	✓	✓	✓	✓	✓	✓	✓
Defending against replay attack		✓	✓	✓	✓	✓	✓

Note. ✓ meet the requirement, do not meet the requirement.

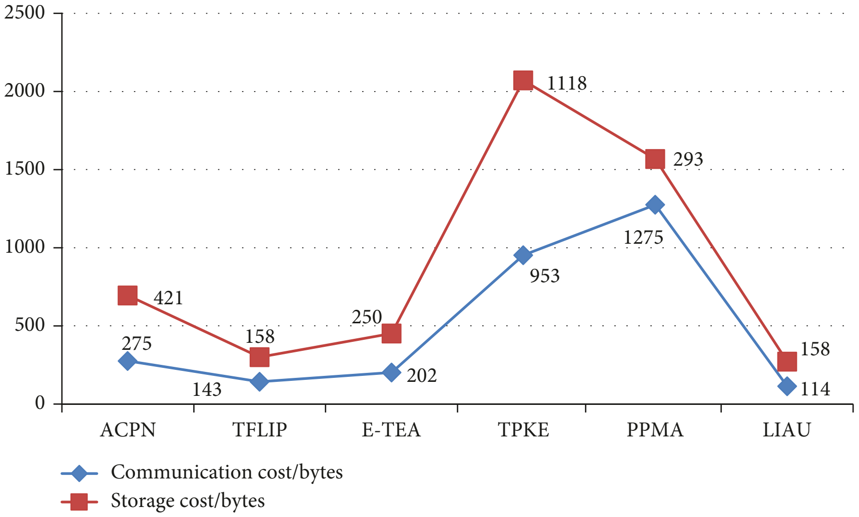


Figure 7: Communication and storage costs.

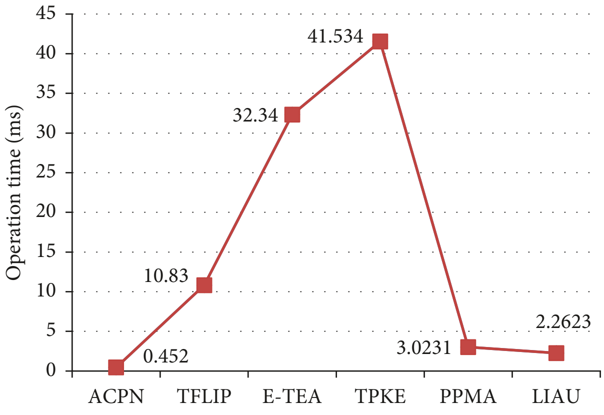


Figure 8: Operation time.

From the perspective of lightweight authentication, we only have analyzed the performance of the LIAU scheme against replay, tampering, and impersonation attack. In fact, the security issue in VANETs is complex and systematic. There are still many problems to be studied and solved.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (61772198), Zhejiang Province Soft Science Foundation of China (2019C35006), and Zhejiang Province Basic Public Welfare Research Project (LGN18F030002).

References

[1] Z. H. Feng, M. Q. Zeng, and J. J. Tao, “Security and privacy issues of 5G VANETs,” *Communication Technology*, vol. 50, no. 5, pp. 1010–1016, 2017.

[2] Y. Xie, L. Wu, Y. Zhang, and L. Ye, “Anonymous Mutual authentication and key agreement protocol in multi-server architecture for VANETs,” *Journal of Computer Research and Development*, vol. 53, no. 10, pp. 2323–2333, 2016.

[3] L. B. Wu, Y. Xie, and Y. B. Zhang, “Efficient and secure message authentication scheme for VANET,” *Journal on Communication*, vol. 37, no. 11, pp. 1–10, 2016.

[4] P. Vijayakumar, M. Azees, A. Kannan, and L. Jegatha Deborah, “Dual authentication and key management techniques for secure data transmission in vehicular ad hoc networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1015–1028, 2016.

[5] M.-C. Chuang and M. C. Chen, “An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics,” *Expert Systems with Applications*, vol. 41, no. 4, pp. 1411–1418, 2014.

[6] S. Kumari, M. Karuppiah, X. Li, F. Wu, A. K. Das, and V. Odelu, “An enhanced and secure trust-extended authentication mechanism for vehicular ad-hoc networks,” *Security*

and Communication Networks, vol. 9, no. 17, pp. 4255–4271, 2016.

[7] J. Li, H. Lu, and M. Guizani, "ACPN: a novel authentication framework with conditional privacy-preservation and non-repudiation for VANETs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 938–948, 2015.

[8] F. Wang, Y. Xu, H. Zhang, Y. Zhang, and L. Zhu, "2FLIP: a two-factor lightweight privacy-preserving authentication scheme for VANET," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 2, pp. 896–911, 2016.

[9] C.-C. Lee and Y.-M. Lai, "Toward a secure batch verification with group testing for VANET," *Wireless Networks*, vol. 19, no. 6, pp. 1441–1449, 2013.

[10] R. Muthumeenakshi, T. R. Reshmi, and K. Murugan, "Extended 3PAKE authentication scheme for value-added services in VANETs," *Computers and Electrical Engineering*, vol. 59, no. 59, pp. 27–38, 2017.

[11] C. Sun, J. Liu, X. Xu, and J. Ma, "A privacy-preserving mutual authentication resisting DoS attacks in VANETs," *IEEE Access*, vol. 5, pp. 24012–24022, 2017.

[12] H. Vasudev and D. Das, "A lightweight Authentication protocol for V2V communication in VANETs," in *Proceedings of the 2018 IEEE SmartWorld, Scalable Computing and Communications*, pp. 1237–1242, Guangzhou, China, October 2018.

[13] S. Ibrahim, M. Hamdy, and E. Shaaban, "Towards an optimum authentication service allocation and availability in VANETs," *International Journal of Network Security*, vol. 19, no. 6, pp. 955–965, 2017.

[14] A. Malik and B. Pandey, "Security analysis of discrete event based threat driven authentication approach in VANET using Petri Nets," *International Journal of Network Security*, vol. 20, no. 4, pp. 601–608, 2018.

[15] M. Bayat, M. Barmshoory, M. Rahimi, and M. R. Aref, "A secure authentication scheme for VANETs with batch verification," *Wireless Networks*, vol. 21, no. 5, pp. 1733–1743, 2015.

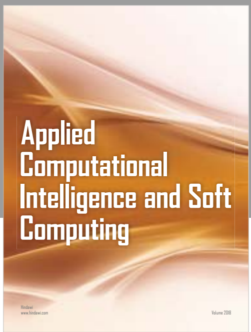
[16] D. Cao and B. Yang, "Design and implementation for MD5-based data integrity checking system," in *Proceedings of the 2010 2nd IEEE International Conference on Information Management and Engineering*, pp. 608–611, Chengdu, China, April 2010.

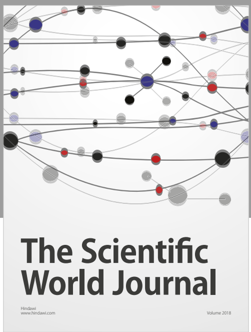
[17] W. Xianquan, "Speed test report of cryptographic algorithm, [EB/OL]," 2011, https://wenku.baidu.com/view/1369a9df6f1aff00bed51e7a.html.

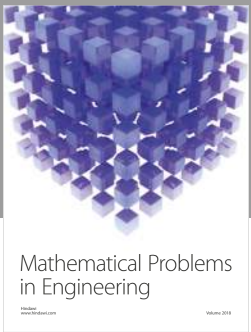
[18] A. Khalid, G. Paul, and A. Chattopadhyay, "RC4-AccSuite: a hardware acceleration suite for RC4-like stream ciphers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 3, pp. 1072–1084, 2017.

[19] K. A. Shim, "An ID-based aggregate signature scheme with constant pairing computations," *Journal of Systems and Software*, vol. 83, no. 10, pp. 1873–1880, 2015.

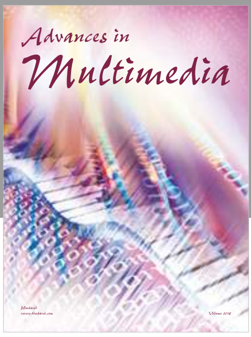
[20] V. Pandi, A. Maria, and V. Chang, "Computationally efficient privacy preserving authentication and key distribution techniques for vehicular ad hoc networks," *Cluster Computing*, vol. 20, pp. 2439–2450, 2017.

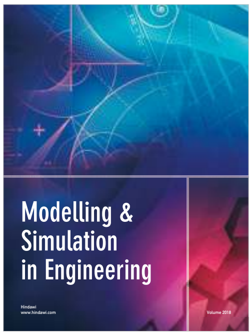

Applied Computational Intelligence and Soft Computing

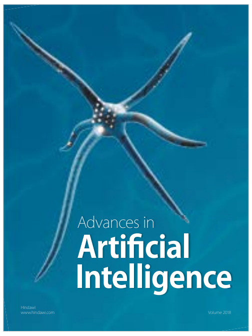

The Scientific World Journal


Mathematical Problems in Engineering

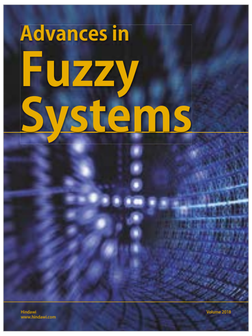
Journal of Engineering

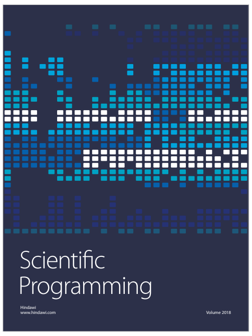

Advances in Multimedia


Modelling & Simulation in Engineering

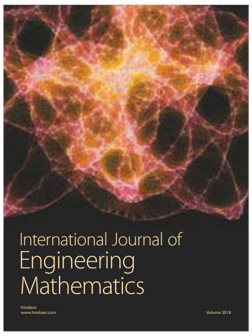

Advances in Artificial Intelligence

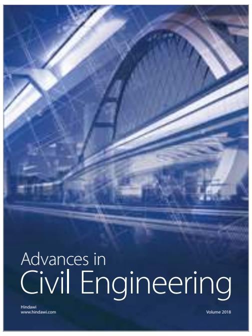
International Journal of Reconfigurable Computing

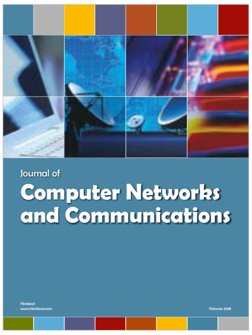

Advances in Fuzzy Systems


Scientific Programming

Advances in Human-Computer Interaction


International Journal of Engineering Mathematics


Advances in Civil Engineering

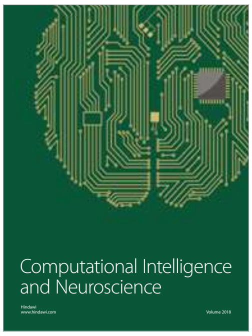

Journal of Computer Networks and Communications

Journal of Robotics

International Journal of Computer Games Technology

International Journal of Biomedical Imaging

Journal of Electrical and Computer Engineering


Computational Intelligence and Neuroscience