Journal of Computer and System Sciences 82 (2016) 1316–1328

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

SEMD: Secure and efficient message dissemination with policy enforcement in VANET ☆

Xuejiao Liu [a], Yingjie Xia [b,*], Wenzhi Chen [b], Yang Xiang [c], Mohammad Mehedi Hassan [d], Abdulhameed Alelaiwi [d]

[a] Institute of Service Engineering, Hangzhou Normal University, Hangzhou, China
[b] College of Computer Science, Zhejiang University, Hangzhou, China
[c] School of Information Technology, Deakin University, Burwood, VIC 3125, Australia
[d] College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

ARTICLE INFO

Article history:
Received 28 August 2015
Received in revised form 2 May 2016
Accepted 10 May 2016
Available online 11 June 2016

Keywords:
Attributed-based encryption
Message dissemination
Outsourcing decryption
VANET

ABSTRACT

Vehicular ad hoc network (VANET) is an increasing important paradigm, which not only provides safety enhancement but also improves roadway system efficiency. However, the security issues of data confidentiality, and access control over transmitted messages in VANET have remained to be solved. In this paper, we propose a secure and efficient message dissemination scheme (SEMD) with policy enforcement in VANET, and construct an outsourcing decryption of ciphertext-policy attribute-based encryption (CP-ABE) to provide differentiated access control services, which makes the vehicles delegate most of the decryption computation to nearest roadside unit (RSU). Performance evaluation demonstrates its efficiency in terms of computational complexity, space complexity, and decryption time. Security proof shows that it is secure against replayable choosen-ciphertext attacks (RCCA) in the standard model.

1. Introduction

In recent years, vehicular ad hoc network (VANET) is regarded as a promising approach for future intelligent transportation system, which enables V2V (vehicles to vehicles) and V2I (vehicles to infrastructure) communication. VANET networks generally consist of On-Board Units (OBU) installed in the vehicles and Road Side Units (RSU) deployed at the roadside, which support cooperative communications of high speed between vehicles and RSUs. Widespread deployment of VANET largely depends on a secure and reliable mechanism for providing accurate data about traffic and road system.

Recently, many security issues have been proposed and addressed to provide security protection of messages in vehicular network [1], in which data confidentiality and access control are the most important. Confidentiality of data ensures that the data is not leaked or disclosed to unauthorized nodes or vehicles. To provide message confidentiality and integrity in VANET communication, encryption needs to be used to allow only the legitimate user to get the message. Traditional symmetrical message encryption would be a solution, but it requires communication cost to establish session keys between message senders and recipients, which greatly slows down the time to access the resource. Another intuitive solution is to encrypt each message with the recipient vehicle's public key, and sign the message before sending. Note that a message is usually

☆ Fully documented templates are available in the elsarticle package on CTAN.

* Corresponding author.
E-mail address: xiayingjie@zju.edu.cn (Y. Xia).

http://dx.doi.org/10.1016/j.jcss.2016.05.006
0022-0000/

sent to several vehicles, there could be many messages ciphertexts under such conventional encryption schemes, which fail to meet real-time requirement of data dissemination in vehicular applications. The issue is extremely serious under the circumstances of safety application's stringent timing constraints. In addition, access control remains a largely unsolved problem, as there is no centralized access control to disseminate the encrypted messages in highly dynamic environment.

To solve the above mentioned issues, we consider the following scenario. In case of an emergency (e.g., a traffic accident, a fire or a serious accident) in a certain area of a city, police headquarters will immediately broadcast emergency messages around. On one hand, they intend to notice policemen, rescue vehicles and etc. nearby to deal with the emergency. On the other hand, they want to exchange information about road conditions to the vehicles which are in this district or will enter this district. Then the receivers can select appropriate routes as soon as possible. The dissemination of these information usually requires a limited range of receivers (e.g., police car or ambulance or the vehicles in a certain area) and have strong real-time requirements (the messages must be processed very quickly). If those alert messages can be accurately disseminated among nearby vehicles in real time, more severe traffic congestions or more serious accidents can probably be avoided to greatly alleviate traffic pressure of the affected area. Thus it is in great need to provide differentiated access services in VANET to guarantee that the messages are delivered to selective vehicles.

Huang et al. [2] is the first one to introduce ciphertext-policy attribute based encryption (CP-ABE) [3] in VANET, in which each vehicle has its capabilities and access rights according to the attributes it owns. Then only the vehicles that have certain attributes satisfying the access policy can decrypt the broadcasted messages with its OBU module. CP-ABE defines access policy by means of attributes, and a vehicle will know whether it can decrypt or not only after decryption attempts by its secret key, and the computation cost in the decryption increases with the number of attributes in the access formula, which usually requires many parings in most of existing CP-ABE schemes. Traditional computers can handle such a task for a typical policy in an acceptable time. However, on OBUs of vehicles, whose processors are often several magnitudes slower than the desktops, there is a significant challenge for these vehicles with resource-limited OBUs to process the messages. Experiments in [4] show that, for an ABE ciphertext containing 100 attributes, the decrypting time required on a mobile terminal device of a high performance machine[1] would be about 30 seconds, following a significant consumption of battery power.

Generally speaking, OBUs have resource-limited processors to make VANETs economically viable [5]. Solutions based on traditional CP-ABE face challenges raised by the limited resource of OBUs. Vehicular network requires advanced cryptography to balance the computation requirements of the algorithm and computation power of OBUs. In this paper, we consider the ABE applications in VANET network in which OBUs are used as information collecting nodes. In order to realize efficient and secure message retrieval at the vehicle, we propose to outsource ABE decryption by generating two security keys, one is an attribute key stored in local RSU, the other one is private key kept secretly in the OBU, then most of the computation in the decryption has been done by the RSU. The vehicle only needs to do little computation, regardless of the complexity of the access policy. In addition, the storage overhead of transformed ciphertext and security key is largely decreased in the OBU. Meanwhile, we introduce a "matching" algorithm before decryption, which can check whether the vehicle can decrypt the message or not without transforming the ciphertext. Our contribution can be addressed in the following aspects.

(1) We propose a secure message dissemination scheme with policy enforcement in VANET network, by encrypting messages using well-defined access policy and then broadcasting them to the related RSUs.

(2) We present a construction of attribute-based encryption with outsourcing decryption algorithm, that enables most of the computation cost to delegate to the nearest RSU without affecting privacy, which largely eliminates the computation overhead for the vehicle, that is constant and efficient regardless of the complexity of the access policy.

Our paper is organized as follows. Section 2 presents some related work about VANET security and CP-ABE. Section 3 introduces the technique preliminaries used in this paper. In Section 4, we present the problem formulation of our system models as well as assumption of our scheme. Detailed SEMD construction is presented in Section 5. In Section 6, we illustrate the performance evaluation and give the security proof of our scheme in detail. Finally, we conclude our paper in Section 7.

2. Related work

Recently, the security issues that threaten the security of VANET have been received considerable attention [1,6,7], and several solutions have been suggested, including vehicular communications security [8], conditional privacy preservation [9–11], authentication [12], OBU key management [5] and etc. In addition, how to assure secure and flexible access control becomes a challenging aspect in the process of message dissemination.

2.1. Message dissemination in VANET

Establishing access control by means of well-known public key infrastructure (PKI) certificates is an effective method. The use of certificates can provide one-to-one message confidentiality between vehicles. However, it can not enforce fine-grained access control for many receivers in dynamic vehicular communication networks.

[1] The mobile device is a 412 MHz iPhone with 128 MB RAM.

Yeh et al. adopted fuzzy identity-based encryption, and presented an attribute-based access control system (ABACS) to provide emergency services, e.g., sending alert messages and selecting rescue vehicles in VANET network [13]. Compared with the existing PKI scheme, they have done extensive studies to show that the computation delay and transmission overhead can be reduced by using attribute-based encryption. However, their schemes do not support complex policy definition with predicates. Huang et al. introduced ciphertext-policy attribute based encryption [3] in VANET to construct an attribute-based security policy enforcement (ASPE) framework [14,2]. Sushmita et al. presented an improved access control scheme in the presence of even one compromised RSU by employing the decentralized attribute based encryption scheme [15].

2.2. Attribute-based encryption

Since its first introduction by Waters et al. [16,3,17], ciphertext-policy attribute-based encryption (CP-ABE) has regarded to be a promising cryptographic tool for access control of encrypted data. It allows data owner to encrypt the data by defining an access policy over attributes, so that only the users associated with a set of attributes which satisfy the policy can decrypt the data. Subsequently, many researchers have studied and proposed a great number of variants of ABE schemes [18–23] in different settings, including non-monotonic access structures, user accountability, attribute revocation, security proof and etc. Nevertheless, almost all of these existing ABE schemes require a large number of exponentiations during decryption. Along with the complexity of access policy, the computational cost in decryption increases linearly, which becomes a bottleneck limiting its application.

To reduce the processing load in decryption with limited resource, some works have been done to improve the decryption efficiency for ABE [24,25,4,26]. Green et al. [4] introduced outsourced decryption of ABE ciphertext and proposed a transformation key transmitted to the proxy by a key blinding technique. And then the proxy can transform ABE ciphertext and do most of the decryption computation by using that key. Zhou et al. [25] presented privacy preserving ciphertext-policy attribute-based encryption (PP-CP-ABE) scheme to protect users' data, and they proposed to outsource some of the encryption and decryption computation to the cloud. However, these works imposed heavy computation or communication burden on the user during data encryption/decryption delegation operation, for example, the set of outsourcing keys were generated and issued by the data owner [24], transformation key (TK) was sent from the user to the proxy each time the user want to decrypt [4]. These schemes are not suitable well in VANET network which introduce extra computation and communication for OBU at the vehicle side.

3. Technical preliminaries

3.1. Pairing-based cryptography

Pairing-based cryptography is the use of a pairing between elements of two groups to a third group with a mapping $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ to construct. If the same group is used for the first two groups (e.g., $\mathbb{G}_1 = \mathbb{G}_2$), the pairing is called symmetric and is a mapping from two elements of one group to an element from a second group. Researches classify pairing instantiations into three basic types: type 1, type 2 and type 3 based on the groups [27].

- Type 1: $\mathbb{G}_1 = \mathbb{G}_2$;
- Type 2: $\mathbb{G}_1 \neq \mathbb{G}_2$ but there is an efficiently computable homomorphism $\phi : \mathbb{G}_2 \to \mathbb{G}_1$;
- Type 3: $\mathbb{G}_1 \neq \mathbb{G}_2$ and there are no efficiently computable homomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$.

The major pairing-based construct is the bilinear map. Type 3 pairings are usually the most efficient, but some authors have preferred type 1 pairings as they are convenient in certain applications. Hence, there are a lot of cryptographic protocols in the literature that have been designed only for type 1 pairings. Therefore, we present our protocol with type 1 pairings in this paper.

3.2. Bilinear maps

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p. Let g be a generator of $\mathbb{G}$ and e be a bilinear map, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with the following properties:

1. Bilinearity: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1$.

If the group operation in $\mathbb{G}$ and the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ are both efficiently computable, then we say $\mathbb{G}$ is a bilinear group.

3.3. Security definition

Definition 1 *(RCCA-secure ABE with outsourcing).* [4] A CP-ABE or KP-ABE scheme with outsourcing is secure against replayable choosen ciphertext attacks [28] if all polynomial time adversaries have at most a negligible advantage in the RCCA game.

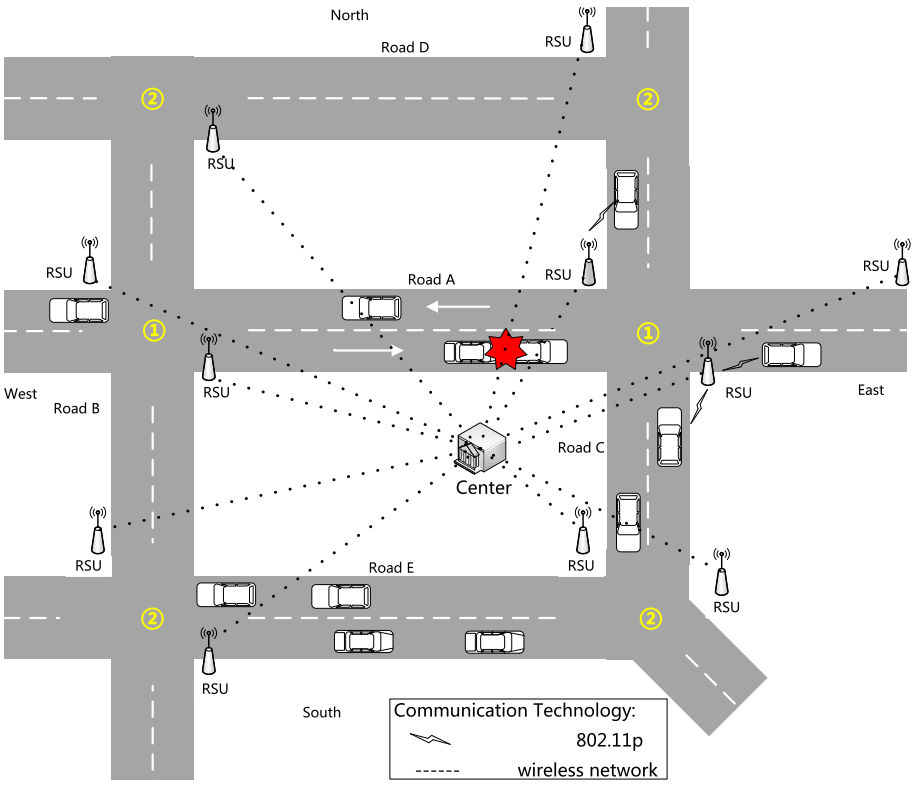

Fig. 1. The sketch map of vehicular network.

From the aspect of security, RCCA is weaker than CCA security, since it considers generating *different* ciphertexts that decrypt to the *same* plaintext as a given ciphertext has the same effect as replaying the same ciphertext several times [28]. That is to say, it allows modifications of the ciphertexts provided they don't change the plaintext.

Definition 2 *(Decisional BDHE [26]).* Let $\mathbb{G}$ be a group of prime order $p \in \Theta(2^{\lambda})$, g is a generator of group $\mathbb{G}$, and $a, s \in \mathbb{Z}_p$ are exponents that be chosen randomly. The decisional q-BDHE assumption is that, given the vector

$$y = \mathbb{G}, p, g, g^s, g^a, ..., g^{(a^q)}, g^{(a^{q+2})}, ..., g^{(a^{2q})}$$

all probabilistic polynomial-time algorithms $\mathcal{A}$ have an advantage negligible in λ of distinguishing $e(g,g)^{a^{q+1}s} \in \mathbb{G}_T$ from a random element in $R \in \mathbb{G}_T$.

The advantage of $\mathcal{A}$ is defined as the probability

$$|Pr[\mathcal{A}(y, e(g,g)^{a^{q+1}s}) = 0] - Pr[\mathcal{A}(y, R) = 0]|$$

in which $a, s \in \mathbb{Z}_p$, $R \in \mathbb{G}_T$, the generator g, and the random bits consumed by $\mathcal{A}$ are all chosen randomly.

Definition 3 *(Selectively secure).* In the selective security game, there will be a stage before init, in which the adversary commits to a challenge access structure M^*. The system is said to be *selectively* secure.

Theorem. *If the above decisional q-BDHE assumption holds, then we can say there is no adversary being able to break the system selectively with a challenge matrix in the selective game.*

4. Approach overview

In this section, we present our system model. Also, we discuss the security threats and requirements in that model.

4.1. System model and assumption

To illustrate our system model, we present a sketch map of vehicular network in Fig. 1, which mainly comprises of three entities: the top trusted center, the road side units RSUs, and the OBUs installed on the vehicles. Vehicles can communicate

with each other and with nearest RSUs using Dedicated Short Range Communication (DSRC) as IEEE 802.11p (5.9 GHz) [29], for range of 300–500M. This communication is an ad hoc communication without wires. The RSUs are deployed across the roads to provide infrastructure support for communication points to and from a WAN (e.g., Internet) in VANET.

The center manages the data analysis operation and controls the message distribution for scientific traffic management, formed by trusted authentication, message encryption and roadway dispatch management module. Trusted authentication module of the center is in charge of the registration of both immobile RSUs and mobile OBUs, and the center is responsible for managing system attributes, generating and distributing security keys according to the vehicles' attributes. RSUs serve as communication relays for vehicles to infrastructure (V2I) communication, which broadcast the messages into the network covering roads areas. Each vehicle is uniquely qualified by a set of identifying attributes which are classified as persistent attributes and dynamic attributes. Persistent attribute values remain constant, e.g., vehicle type, color, brand, and etc. Dynamic attribute values change frequently, such as speed, direction, location of the vehicle, the road and etc. We assume that each vehicle has devices known as tamper resistance to store private keys and other critical data, as employed in [14]. We identify the streets as 1st or 2nd and etc. street segment from the affected street due to the emergency.

We consider that center is a fully trusted organization and RSUs are honest-but-curious, that is to say, the RSUs honestly carry out the computations delegated by the vehicle, and curiously infer additional privacy information, but they can not recover the encrypted data by what they has.

4.2. Security requirements

In order to guarantee the message disseminated to selective receivers, we describe our threat model as well as security requirements in the following.

An attacker may try to obtain the policy encrypted messages by injecting, altering and replaying them once after the message comes out. Also the attacker may collude with other vehicles to access encrypted messages. We try to achieve the following security requirements.

(1) Data confidentiality: Confidentiality is one of the major security requirements in VANET [30]. The vehicles which do not have security keys satisfying the access policy of the messages must be prevented from accessing in VANET.

(2) Policy enforcement: Messages should be enforced with access policy in the ciphertexts and delivered selectively to the vehicles according to the policy, without disclosing any content of the policy and the message.

(3) Efficient: To ensure efficient VANET computation at the vehicle, the necessary cryptographic operations for OBUs should be lightweight and it should not introduce significant communication overhead, as they possess resource-limited processors.

5. Our proposed scheme

In this section, we will give our algorithm construction in detail and present our SEMD scheme of that construction in specific application.

5.1. Algorithm construction

In order to reduce the computation cost for the user, we propose an improved CP-ABE algorithm which can delegate most of the decryption computation. Our construction mainly comprises of the following five parts.

`Setup`(U): In the setup step, it takes security parameter and the attributes set U of the system as input, and takes public key and master secret key as output. The algorithm first chooses a group $\mathbb{G}$ with prime order p and a generator g of that group. Also it chooses a great number of group elements $h_1, ..., h_U \in \mathbb{G}$ associated with each attribute of the attributes set U at random. Besides, the system chooses two exponents in $\mathbb{Z}_p$ at random, that is $\alpha_1 \in \mathbb{Z}_p$, $\alpha_2 \in \mathbb{Z}_p$, and let $\alpha = (\alpha_1 + \alpha_2) \bmod p$. In addition, it chooses random exponent $a \in \mathbb{Z}_p$. Finally, the public key PK is published as

$$g, e(g,g)^{\alpha}, g^{a}, h_1, ..., h_U$$

The master secret key is set to be MK = g^{α}.

`Encrypt`(PK, (M, ρ), m): In the encryption step, the algorithm will certainly takes a message m to encrypt as input. Also, it should take the public key PK and an access structure to encrypt the message as input, and finally output the corresponding ciphertext. In our construction, the algorithm takes a linear secret sharing scheme (LSSS) to define the access matrix (M, ρ), in which M is defined as an $l \times n$ matrix, and ρ refers to the function that maps rows of M to the attributes. The algorithm first chooses a random vector $v = (s, y_2, ..., y_n)^T \in \mathbb{Z}_p^n$, in which the exponent $s \in \mathbb{Z}_p$ is randomly chosen as the secret to be shared. The other values are used to share the encryption exponent s. For $i = 1$ to l, it calculates $\lambda_i = M_i v$, in which M_i is the vector associated with the ith row of M. Also it should choose several random exponents $r_1, ..., r_l \in \mathbb{Z}_p$ in the encryption calculation. The ciphertext CT is generated as

$$C = me(g,g)^{\alpha s}, \quad C' = g^{s},$$
$$(C_1 = g^{a\lambda_1} h_{\rho(1)}^{-r_1}, D_1 = g^{r_1}), ..., C_l = g^{a\lambda_l} h_{\rho(l)}^{-r_l}, D_l = g^{r_l})$$

Table 1
Notations in our algorithm construction.

U	an attribute universe description
$h_1, ..., h_U$	group elements associated with the attributes set U
PK,MK	public parameter key and system master key
α	master secret key component
S	a set of attributes
SK,AK	private key and attributes key for user
M	an $l \times n$ matrix
ρ	a function that maps rows of M to attributes
s	a random parameter secret to be shared
w_i	a set of constants
λ_i	valid shares of the secret s
CT, CT′	ciphertext, and partially decrypted ciphertext
m	message

`Key Generate`(MK,S): In the key generation step, the algorithm takes the master secret key and a set of attributes related to a user as input. The algorithm firstly chooses a random number $t \in \mathbb{Z}_p$. It creates the private key as two parts, one part is AK, that is "attribute key" for the proxy, and the other one is SK, that is private "security key" for the users.

$$\text{AK} : (K_1 = g^{\alpha_1} g^{at}, L = g^t, \forall x \in S : K_x = h_x^t)$$
$$\text{SK} : (K_2 = g^{\alpha_2} g^{at})$$

`Transform Ciphertext`(CT,AK): This algorithm takes a ciphertext CT for access structure (M, ρ) and an attribute key AK for a set of attributes S as input. Suppose that S is a set of attributes that satisfy the access structure (M, ρ), which means an authorized user. We denote $I \subset \{1, 2, ..., l\}$ as $I = \{i, \rho(i) \in S\}$ and define $\{w_i \in \mathbb{Z}_p\}_{i \in I}$ as a set of constants. According to the access structure of M, only if $\sum_{i \in I} w_i \lambda_i = s$ holds, then we can say λ_i are valid shares of the secret s. (Note there could potentially be different ways of choosing the w_i values to satisfy this.)

1. In terms of the following equation, the proxy checks whether the users can decrypt the ciphertext.

$$\sum_{i \in I} M_i w_i = (1, 0, ..0)$$

2. The proxy transforms the ciphertext CT into transformed ciphertext CT′.

$$\begin{aligned} \text{CT}' &= e(C', K_1)/(\prod_{i \in I}(e(C_i, L)e(D_i, K_{\rho(i)}))^{w_i})^2 \\ &= e(g^s, g^{\alpha_1} g^{at})/(e(g, g)^{ats})^2 \\ &= e(g, g)^{\alpha_1 s}/e(g, g)^{ats} \end{aligned}$$

The proxy will then send CT′ to users.

`Decrypt`(CT′,SK): This algorithm takes transformed ciphertext CT′ and user's private key SK as input, and then takes the plaintext m as output.

$$\begin{aligned} e(\text{SK}, C')\text{CT}' &= e(g^{\alpha_2} g^{at}, g^s)e(g, g)^{\alpha_1 s}/e(g, g)^{ats} \\ &= e(g, g)^{\alpha_2 s} e(g, g)^{ats} e(g, g)^{\alpha_1 s}/e(g, g)^{ats} \\ &= e(g, g)^{\alpha s} \end{aligned}$$

The plaintext m will be got by $m = C/e(g, g)^{\alpha s}$. This kind of decryption only requires one pairing computation. The scheme not only guarantees data security, but also achieves fast decryption at the user side. Most of the notations used in our scheme are summarized in Table 1.

5.2. Scheme application

The center runs `Setup` algorithm to produce the system public parameters PK and master key MK. Our proposed SEMD scheme mainly consists of three phases: (1) vehicle attestation and registration; (2) message dissemination with policy enforcement; (3) message decryption.

Vehicle attestation and registration When vehicles move across RSUs, they need to register at a nearby RSU. Before registering, the vehicles firstly get their OBU modules attested by the RSU, to verify whether it is an accurate and valid device. After successful attestation, the RSU will obtain and record the vehicles' attributes values.

Algorithm 1 OBU attestation and registration.

1: Vehicle v moves across RSU
2: v broadcasts its encrypted license plate number: $(LN_v)_{PK_{RSU}}$
3: RSU gets LN_v by SK_{RSU}
4: **if** $LN_v \in$ DMV **then**
5: RSU retrieves persistent attributes: $\{type, color, year, ...\}_v$
6: **end if**
7: v sends dynamic attributes: $\{speed, loc, dir, ...\}_v$ to RSU
8: RSU transfers these attributes to Center
9: Center runs `Key Generate` to generate AK and SK
10: Center sends AK $\rightarrow$ RSU, SK $\rightarrow$ v

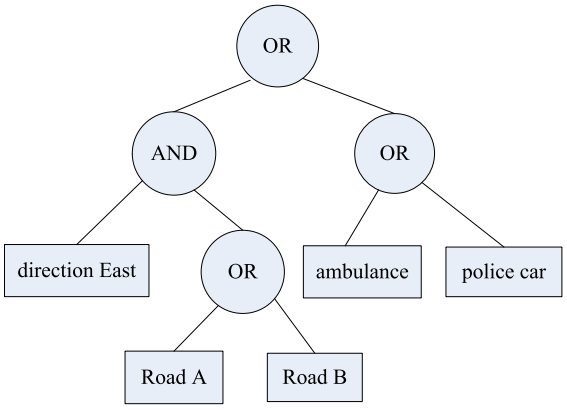

Fig. 2. Access policy of an alert message.

As shown in Algorithm 1, upon having verified the validity of the OBU, the vehicle registers its attribute values with the local RSU. The vehicle v broadcasts a registration request by specifying the vehicle's license plate number, LN_v, encrypted using the public key of the RSU PK_{RSU}. The RSU, upon receiving the request, verifies the license plate number from vehicle databases such as DMV (Motor Vehicle Database), and retrieves all the persistent attributes of the vehicle. Then it requests the vehicle for dynamic attribute values, that is, its current speed, location and direction values and etc. The RSU employs location proof methodologies to verify location-related attributes [31]. The registration ends with the RSU sending a message with these attributes of the vehicle to the center. Then the center runs `Key Generate` algorithm using these attributes to generate two keys for the vehicle, one part is concealed in what we called an "attribute key" AK, and is sent to the RSU. The other one is hidden in "security key" SK which must be kept private by the vehicles.

A vehicle registers at the nearest RSU at the start of its trip. When the vehicle leaves the RSU, the RSU will forward the vehicles' information to the next RSU located in the vehicle's direction in a secure channel. If the vehicle has changed its direction, it will register to the next nearest RSU again.

Message dissemination with policy enforcement As shown in Fig. 1, upon receiving an event report (e.g., traffic accident) forwarded by any vehicle or RSU, the center immediately launches the emergency plan. Firstly, it verifies and checks the status of the emergency. Then it would generate an alert message to the related vehicles and a rescue message transferred to the nearest rescue cars. To establish an efficient message dissemination way, these messages should be exchanged not only based on the vehicles' functions (e.g., police cars, ambulances, commercial vehicles and etc.), but also on vehicles' movements, such as location, direction and etc. These applications need selective message delivery approach to perform access control on messages disseminated in VANET.

According to the severity of the emergency and the estimated time to solve it, the center decides to broadcast the messages in a certain distance (e.g., to the 2nd streets or 3rd streets from the accident as illustrated in Fig. 1) for a period of time. Specifying a policy is to define a collection with multiple attributes and logical predicates, to ensure data access by authorized vehicles. In our scheme, the center could define access policy, which typically involves constraints on the attribute values of potential recipient vehicles. We present an access tree in Fig. 2 to illustrate a cryptography-binding policy group for message dissemination in Fig. 1.

The center runs `Encrypt` algorithm to encrypt the message m with access policy and takes a ciphertext CT as output. It assures that only the vehicles with a set of attributes that satisfy the policy can be able to decrypt the message. Then the center disseminates the message to the corresponding RSUs near the area with internet, and RSUs broadcast the messages via DSRC.

Message decryption Since the attribute keys of registered vehicles are stored in the RSU, the RSU will perform fast decryption test, to decide whether the attributes set in vehicle's AK match access policy in ciphertexts without decryption. If the test returns false, the RSU will not deliver the messages to the vehicles. Otherwise, the RSU will perform `Transform Ciphertext`(CT,AK) algorithm, which takes in a ciphertext CT and an AK corresponding to vehicles' attributes S, and

Table 2
Computational complexity.

Operation	Complexity
Setup (center)	$O(1)$
Encrypt (center)	$O(2lT_e)$
Key Generate (center)	$O((\|S\|+3)T_e)$
Transform Ciphertext (RSU)	$O((2+l)T_p)$
Decrypt Ciphertext (OBU)	$O(T_p)$

$|S|$ means the number of attributes in the set S for the vehicle.
l is the row of matrix M, means the number of attributes in the access policy.

Table 3
Comparison with computation cost.

Scheme	Encryption		Decryption	
	T_e	T_p	T_e	T_p
ASPE [2]	$2l+4$	–	l	$2l+2$
IACM [15]	$5l+1$	1	l	$2l$
ESAC [32]	$3l$	–	–	2
Ours	$3l$	–	–	1

transforms the ciphertext into CT′. Finally the RSU sends CT′ to the responding vehicles. While the vehicles only need to do little decryption computation by its own secret key, which is quite efficient.

6. Performance analysis

In order to illustrate the efficiency of our scheme, in this section we first present the experimental setup in our experiments, then provide the performance evaluation, including computational cost, communication cost and decryption efficiency in our scheme. Meanwhile, theoretical analysis and experimental results demonstrate the decryption efficiency of our proposed protocol in comparison with existing schemes. Finally, we give the formal security proof of our scheme.

6.1. Experimental setup

In order to evaluate the performance of our scheme, we simulate message dissemination application in VANET in the following platforms. We deploy the center in trusted Aliyun cloud server, and we assume RSUs have abundant computational resources and storage (e.g., with several GB of RAMs and GHz processors). Generally speaking, the OBU has limited computation power, such as a 400 MHz processor employed in [5]. We have done a set of experiments on three different dedicated platforms as vehicles' OBU to show the efficiency advantage. They are a 3.20 GHz Intel Core CPU with 4 GB of RAM running 32-bit Linux Kernel version 3.2.0 (denoted as Intel), a 1.3 GHz ARM-based Nexus ME370T with 1 GB of RAM running Android OS (denoted as Pad), and a 1536 MHz ARM-based HTC G18 with 768 MB of RAM running Android OS (denoted as Phone).

6.2. Computational complexity

In this section, we discuss the theoretical performance of our scheme. We discuss the computation overheads involved in the proposed protocol and compare it with the existing schemes. According to the algorithm construction in section 5, the computational complexities of all steps in our scheme are summarized in Table 2. As pairing computations and exponentiations are most expensive operations, we calculate the number of pairing operations and the number of exponentiations performed in the algorithms. T_p denotes the computation cost of pairing, T_e denotes the computation cost of exponentiation.

There is no computationally intensive task involved during the key issuing and message encryption stage for the center. In the decryption phase, the bulk of decryption operation is now handled by the RSU, which substantially reduces the decryption time required for vehicles to recover the plaintext.

Comparing with the existing schemes in VANET, we calculate the computational complexity in the aspects of encryption and decryption of OBU module in Table 3. We denote attribute-based secure policy enforcement scheme in [2] as ASPE, and improved access control mechanism scheme in [15] as IACM. The ESAC in [32] is short for computationally efficient secure access control in vehicular ad hoc network.

The encryption algorithm will require three exponentiations in $\mathbb{G}$ to compute C_i and D_i for each row in the ciphertext access policy. Thus, the total computation cost of encryption is $3l$ exponentiations, in which $2l$ and l in C_i and D_i, respectively. In its simplest form, the decryption algorithm requires only one pairing operations for $e(\mathrm{SK}, C')$. While in [2] and [15], the number of pairing computations grows linearly with the number of attributes required for decryption, and there are

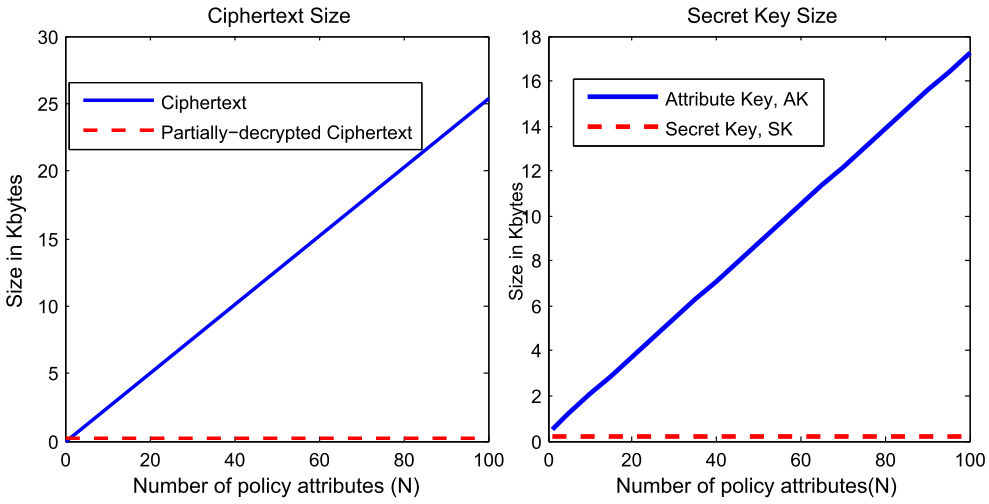

Fig. 3. The size of ciphertext and secret key in the vehicle.

two pairing computations in [32], whereas our scheme requires only one paring computations during decryption. Therefore, we see that our scheme is more efficient for resource-limited OBU module in vehicular ad hoc networks.

6.3. Space complexity

We use libfenc library [33], which uses key encapsulation mechanism, and adopts elliptic curve from Pairing-Based Crypto library of Stanford [34] to implement our scheme in software.

In traditional CP-ABE scheme, both of the decryption time and ciphertext size depend on the complexity of the access policy attached in the ciphertext. Moreover they grow linearly with the increasing number of attributes in that policy. To illustrate this, we choose 100 of the most complex policies as the form (A_1 AND A_2 AND...AND A_n), of which A_i is an attribute, and the values of N increase from 1 to 100 in our experiments. This approach ensures that all the ciphertext components are involved in the decryption phrase. In each case, we construct a standard decryption key that contains exact N attributes. In VANET network, the number of attributes may not increase to 100 attributes in a short time in the policy, but the number could grow with the complex policy.

In practice, ABE is a pubic key encryption mechanism not suitable for encrypting data directly. In our experiment, the message is encrypted separately using a symmetric encryption scheme AES (Advanced Encryption Standard) under a symmetric key k and the ABE ciphertext is the encryption of that symmetric key k. As existing works in [17,13,4,2], here we do not consider the influence of symmetric encryption k as consideration.

Recall that in our scheme, we outsource the partial decryption computation by generating an attribute key AK and applying the `Transform Ciphertext` algorithm to the ABE ciphertext using this key. As expected in Fig. 3, an encryption under a ciphertext policy with 100 attributes results in the ciphertext of nearly 25.4 KB. However, in our implementation each partially-decrypted ciphertext with a constant size of 176 bytes, and each security key for the vehicle with 168 bytes, regardless of the complexity of its corresponding ciphertext policy. Instead of storing the whole ciphertext and private key, the vehicle takes very less space to store the partially decrypted ciphertext and security key. Since each OBU has limited memory space, storing limited data in this memory would be preferable.

6.4. Efficiency

Fig. 4 shows the measured decryption times on the three test platforms with outsourcing and non-outsourcing, as a function of policy attribute N.

To reduce the effect of experimental variability, we do our experiment several times for each ciphertext policy and we take the average values as the results illustrated in the following figures. Without outsourcing, we can see that it takes about 1.47 seconds for the Intel platform to decrypt the ciphertext under a ciphertext policy with 100 attributes. On the other hand, decryption time degrades considerably on the phone platform: it requires almost 12.6 seconds under a policy with 100 attributes. For pad, it costs 10.3 seconds. By outsourcing the decryption of ABE ciphertext using our scheme, the final decryption requires only 11 milliseconds on the desktop, about 93 milliseconds on the phone platform, and 68 milliseconds on the pad platform. And the time remains constant regardless of the complexity of access policy.

Recall that in our scheme, we delegate the most complex decryption calculation to RSUs. The average transformed decryption time by RSU is less than the time on Intel with non-outsource scheme in Fig. 4. Also, we add a matching operation before the RSU's transforming of the ciphertext, which used to check whether the user can decrypt the ciphertext

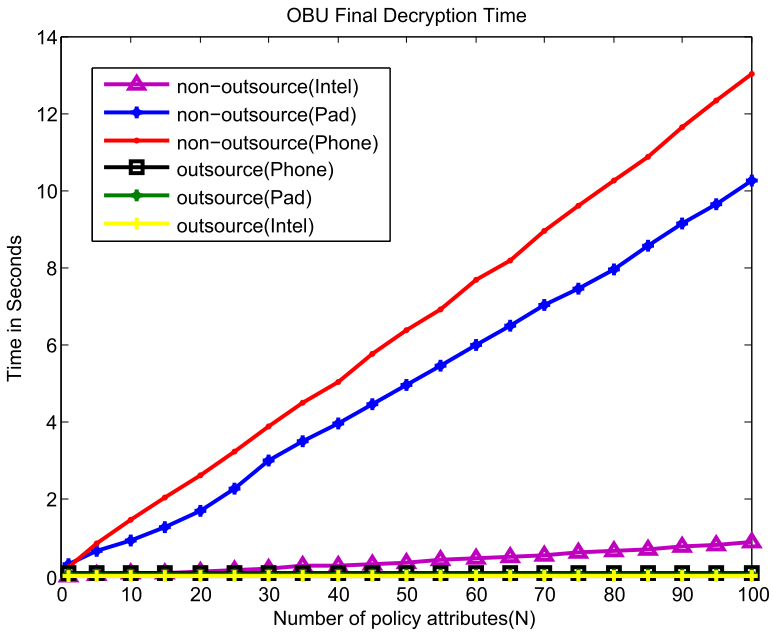

Fig. 4. The final decryption time of OBU in different configurations with our scheme and non-outsourcing scheme.

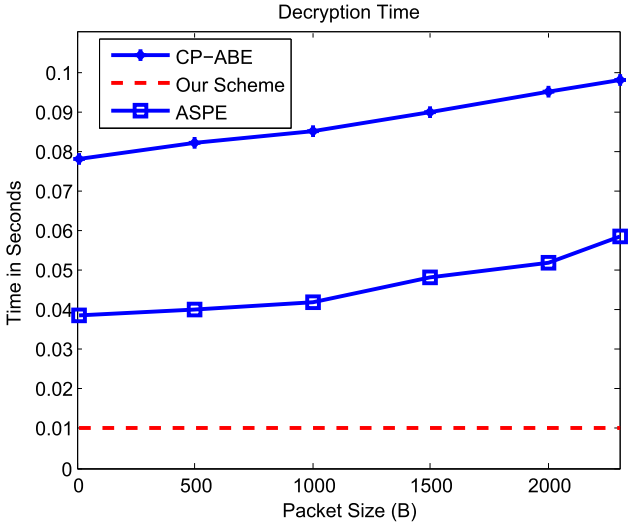

Fig. 5. The comparison of decryption time with the increase of packet size.

or not. It saves a lot of calculation time because if it is not satisfied, then the RSU will not transform the corresponding ciphertext. As a result, this process shortens the whole transformed decryption time for the RSU. Generally speaking, the average transform decryption time of RSU is linearly increasing with the number of vehicles in the density. The more the number of vehicles is, the more transform decryption time it will take. In the coverage range of DSRC system, the RSU will take at most 2.5 s transform decryption time and 1.3 s average transformation decryption time in the case of 30 vehicles. And the time could be accelerated with high performance RSU devices.

6.5. Comparison

Compared with the CP-ABE implementation of traditional CP-ABE scheme provided in [35] and the existing ASPE scheme in [2], we give a comparison of decryption time with the size of packet payload in Fig. 5. We assume that V2V communication uses 802.11 based technologies and V2I uses DSRC technology [36]. Following 802.11 standards, the maximum allowable payload size is 2312 bytes, so we use in our simulation to provide the worst case performance results. For traditional CP-ABE and ASPE scheme, their experiments have been conducted on a 64-bit platform with a 3.2 GHz processor.

We have done our scheme on a 32-bit Intel Core with 3.20 GHz CPU and 4 GB RAM. It can be observed that, with the increasing size of packets, the decryption time in ASPE is increasing along with the packet size, while the time is constant in our scheme. As is shown in Fig. 5, for a packet size of 2300 bytes, the decryption time for CP-ABE is 0.098 s, it is 0.052 s for ASPE scheme, whereas in our scheme the time is only 0.01 s at the vehicle side. It is because we outsource the most complex decryption calculation to the RSU and leave only one pairing to the vehicle. Thus, it can be proved that our scheme outperforms their schemes in that it can enormously reduce decryption time at the vehicle side.

6.6. Security proof

Suppose we have an adversary $\mathcal{A}$, which chooses a challenge matrix M^* of size $l^* \times n^*$ ($l^*, n^* \leq q$). That is to say, both dimensions of the matrix are at most q. In the following we present how to play the decisional q-BDHE problem with a simulator $\mathcal{B}$. The adversary has non-negligible advantage $\varepsilon = \mathrm{Adv}_{\mathcal{A}}$ against our construction.

Init The simulator inputs a q-BDHE challenge $\overrightarrow{y} = (g, g^s, g^a, ..., g^{a^q}, g^{a^{q+2}}, ..., g^{a^{2q}})$, T. The adversary gives him the challenge access structure (M^*, ρ^*), in which M^* has $n^* \leq q$ columns.

Setup The simulator chooses $\alpha'_1, \alpha'_2 \in \mathbb{Z}_p$ at random and implicitly sets $\alpha_1 = \alpha'_1 + a^{q+1}/2$, $\alpha_2 = \alpha'_2 + a^{q+1}/2$, by letting $e(g,g)^{\alpha_1} = e(g^{a/2}, g^{a^q/2}) e(g,g)^{\alpha'_1}$, $e(g,g)^{\alpha_2} = e(g^{a/2}, g^{a^q/2}) e(g,g)^{\alpha'_2}$.

To "program" the group elements $h_1, ..., h_U$, the simulator $\mathcal{B}$ selects a value $z_x \in \mathbb{Z}_p$ at random for each x in the attributes set U. If the equation $\rho^*(i) = x$ holds, the simulator programs h_x as:

$h_x = g^{z_x} \prod_{i \in X} g^{a M^*_{i,1}} g^{a^2 M^*_{i,2}} \cdots g^{a^{n^*} M^*_{i,n}}$

Note that X denotes the set of indices i and if $X = \Phi$ then $h_x = g^{z_x}$.

Here we point out several points. Firstly, due to the g^{z_x} parameters, the values of h_x are distributed randomly. Also, for each x in the attributes set, ρ^* is an injective function that means there is at most one i to satisfy $\rho^*(i) = x$.

Then the adversary sets the public parameters PK as $\{g, e(g,g)^{\alpha}, g^a, h_1, ..., h_U\}$ and sends them to simulator.

Phase 1 In this phase, the simulator $\mathcal{B}$ answers private key queries. There is a set of attributes S that does not satisfy the access structure M^*. Suppose that the simulator $\mathcal{B}$ is given a security key query for the set S.

The simulator first chooses a random $r \in \mathbb{Z}_p$. Then it finds a vector $\overrightarrow{w} = (w_1, ..., w^*_n) \in \mathbb{Z}_p^{n^*}$ such that $w_1 = -1$ and for all i where $\rho^*(i) \in S$ we have that $\overrightarrow{w} \cdot M^*_i = 0$. According to linear secret sharing scheme (LSSS), since S does not satisfy M^*, such a vector must exist. At first, the simulator implicitly defines t as:

$$r + w_1 a^q + w_2 a^{q-1} + \cdots + w_{n^*} a^{q-n^*+1}$$

Then it performs

$$L = g^r \prod_{i=1,...,n^*} (g^{a^{q+1-i}})^{w_i} = g^t.$$

By the definition of t, the unknown term in g^{α} can be ignored in the process of creating private keys. The simulator can compute K_1, K_2 as:

$$K_1 = g^{\alpha_1} g^{ar} \prod_{i=2,...,n^*} (g^{a^{q+2-i}})^{w_i}$$

$$K_2 = g^{\alpha_2} g^{ar} \prod_{i=2,...,n^*} (g^{a^{q+2-i}})^{w_i}$$

Now for $x \in S$ we calculate K_x. First, we consider $x \in S$ for which there is no i such that $\rho^*(i) = x$. For those we can simply let $K_x = L^{z_x}$. Here the more challenging task is to get key components K_x for the attributes x used in the access structure. Note that we can not simulate for those keys because of no terms of the form $g^{a^{q+1}}$. In the process of calculating h^t_x, all terms of the exponent come from $M^*_{i,j} a^j w_j a^{q+1-j}$ for some j, where $\rho^*(i) = x$. However, when combined everything with an exponent of a^{q+1}, we have that $M^*_i \cdot \overrightarrow{w} = 0$ cancels.

The simulator creates K_x in the following. Suppose $\rho^*(i) = x$, then

$$K_x = L^{z_x} \prod_{j=1,...,n^*} (g^{a^j r} \prod_{k=1,...,n^*, k \neq j} (g^{a^{q+1+j-k}})^{w_k})^{M^*_{i,j}}$$

Challenge The adversary gives two messages m_0, m_1 to the simulator. The simulator flips a coin β. It creates $C = m_\beta T e(g^s, g^{\alpha'_1}) e(g^s, g^{\alpha'_2})$ and $C' = g^s$. Since the term $h^s_{\rho^*(i)}$ will contain terms of $g^{a^j s}$, the tricky part is how to simulate C_i values. The simulator can use the secret splitting to cancel out these parts. Intuitively, the simulator chooses random $y'_2, ..., y'_{n^*} \in \mathbb{Z}_p$ and then uses the $\overrightarrow{v} = (s, sa + y'_2, ..., sa^{n^*-1} + y'_{n^*}) \in \mathbb{Z}_p^{n^*}$ to share the secret.

Besides, the simulator chooses values $r'_1, ..., r'_l$ at random.

For $i = 1, ..., n^*$, the ciphertext components are generated as $D_i = g^{-r'_i} \cdot g^{-s} = g^{-r'_i - s}$, $C_i = h^{-r'_i}_{\rho(i)} \cdot (\prod_{i=2,...,n^*} (g^a)^{M^*_{i,j} \cdot y'_j} \cdot (g^s)^{-z_{\rho^*(i)}})$

Phase 2 The simulator $\mathcal{B}$ responds to $\mathcal{A}$'s queries in the same manner as phase 1, except that it refuses to answer any query that would result in a set S that satisfy M^*.

Guess Finally the adversary will output a guess β' of β. The simulator will also give its guess. It outputs 0 to show that it guesses $T = e(g,g)^{a^{q+1}s}$ if $\beta = \beta'$; otherwise, it outputs 1 to indicate that it believes T is a random group element in G_T. The simulator $\mathcal{B}$ gives the final simulation

$$Pr[\mathcal{B}(\overrightarrow{y}, T = e(g,g)^{a^{q+1}s}) = 0] = 1/2 + Adv_{\mathcal{A}}$$

If T is a random group element, the message m_β is entirely hidden. So that the probability is $Pr[\mathcal{B}(\overrightarrow{y}, T = e(g,g)^{a^{q+1}s}) = 0] = 1/2$. Thus, the simulator $\mathcal{B}$ plays the decisional q-DBHE game with non-negligible advantage.

7. Conclusion

In this paper, we present a secure and efficient message dissemination scheme with policy enforcement to provide data confidentiality and differential access service of message in vehicular network. Performance analysis demonstrates that our scheme not only provides fast decryption, but also reduces storage overhead at the vehicle side. Security analysis shows that the proposed construction is proven to be RCCA secure in the standard model. In the future, we would like to refine this scheme, such as efficient authentication and revocation of OBU in vehicular network.

Acknowledgments

This research is supported in part by the following funds: National Natural Science Foundation of China under grant number 61472113, 61304188 and 61502134, Zhejiang Provincial Natural Science Foundation of China under grant number LZ13F020004 and LR14F020003, and Zhejiang Provincial Science and Technology Innovation Program under grant number 2013TD03.

References

[1] R. Di Pietro, S. Guarino, N. Verde, J. Domingo-Ferrer, Security in wireless ad-hoc networks–a survey, Comput. Commun. 51 (2014) 1–20.
[2] D. Huang, M. Verma, Aspe: attribute-based secure policy enforcement in vehicular ad hoc networks, Ad Hoc Netw. 7 (8) (2009) 1526–1535.
[3] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in: IEEE Symposium on Security and Privacy, 2007, pp. 321–334.
[4] M. Green, S. Hohenberger, B. Waters, Outsourcing the decryption of ABE ciphertexts, in: Proceedings of the 20th Usenix Conference on Security, 2011, pp. 1–16.
[5] A. Studer, E. Shi, F. Bai, A. Perrig, Tacking together efficient authentication, revocation, and privacy in vanets, in: Proceedings of the 6th Annual IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks, IEEE, 2009, pp. 1–9.
[6] R.G. Engoulou, M. Bellaïche, S. Pierre, A. Quintero, Vanet security surveys, Comput. Commun. 44 (2014) 1–13.
[7] M.N. Mejri, J. Ben-Othman, M. Hamdi, Survey on vanet security challenges and possible cryptographic solutions, Veh. Commun. 1 (2) (2014) 53–66.
[8] X. Sun, X. Lin, P.-H. Ho, Secure vehicular communications based on group signature and id-based signature scheme, in: IEEE International Conference on Communications, IEEE, 2007, pp. 1539–1545.
[9] R. Lu, X. Lin, H. Zhu, P.-H. Ho, X. Shen, Ecpp: efficient conditional privacy preservation protocol for secure vehicular communications, in: Proceedings of the 27th Conference on Computer Communications, IEEE, 2008, pp. 1229–1237.
[10] D. Huang, S. Misra, M. Verma, G. Xue, Pacp: an efficient pseudonymous authentication-based conditional privacy protocol for vanets, IEEE Trans. Intell. Transp. Syst. 12 (3) (2011) 736–746.
[11] T. Chim, S. Yiu, L. Hui, V. Li Vspn, Vanet-based secure and privacy-preserving navigation, IEEE Trans. Comput. 63 (2) (2014) 510–524.
[12] J.K. Liu, T.H. Yuen, M.H. Au, W. Susilo, Improvements on an authentication scheme for vehicular sensor networks, Expert Syst. Appl. 41 (5) (2014) 2559–2564.
[13] L.-Y. Yeh, Y.-C. Chen, J.-L. Huang, Abacs: an attribute-based access control system for emergency services over vehicular ad hoc networks, IEEE J. Sel. Areas Commun. 29 (3) (2011) 630–643.
[14] X. Hong, D. Huang, M. Gerla, Z. Cao, Sat: situation-aware trust architecture for vehicular networks, in: Proceedings of the 3rd International Workshop on Mobility in the Evolving Internet Architecture, ACM, 2008, pp. 31–36.
[15] S. Ruj, A. Nayak, I. Stojmenovic, Improved access control mechanism in vehicular ad hoc networks, in: Ad-hoc, Mobile, and Wireless Networks, Springer, 2011, pp. 191–205.
[16] A. Sahai, B. Waters, Fuzzy identity-based encryption, in: Advances in Cryptology-Eurocrypto, 2005, pp. 457–473.
[17] B. Waters, Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization, in: Public Key Cryptography, Springer, 2011, pp. 53–70.
[18] L. Cheung, C. Newport, Provably secure ciphertext policy ABE, in: Proceedings of the 14th ACM Conference on Computer and Communications Security, 2007, pp. 456–465.
[19] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters, Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption, in: Advances in Cryptology-Eurocrypto, 2010, pp. 62–91.
[20] J. Li, K. Ren, B. Zhu, Z. Wan, Privacy-aware attribute-based encryption with user accountability, in: Information Security, 2009, pp. 347–362.
[21] R. Ostrovsky, A. Sahai, B. Waters, Attribute-based encryption with non-monotonic access structures, in: Proceedings of the 14th ACM Conference on Computer and Communications Security, 2007, pp. 195–203.
[22] A. Sahai, H. Seyalioglu, B. Waters, Dynamic credentials and ciphertext delegation for attribute-based encryption, in: Advances in Cryptology, 2012, pp. 199–217.
[23] B. Waters, Dual system encryption: realizing fully secure ibe and hibe under simple assumptions, in: Advances in Cryptology, 2009, pp. 619–636.
[24] J. Li, C. Jia, J. Li, X. Chen, Outsourcing encryption of attribute-based encryption with mapreduce, in: Information and Communications Security, Springer, 2012, pp. 191–201.
[25] Z. Zhou, D. Huang, Efficient and secure data storage operations for mobile cloud computing, in: Proceedings of the 8th International Conference on Network and Service Management, 2012, pp. 37–45.

[26] S. Hohenberger, B. Waters, Attribute-based encryption with fast decryption, in: Public Key Cryptography, 2013, pp. 162–179.
[27] N. Koblitz, A. Menezes, Pairing-Based Cryptography at High Security Levels, Springer, 2005.
[28] R. Canetti, H. Krawczyk, J.B. Nielsen, Relaxing chosen-ciphertext security, in: Advances in Cryptology, 2003, pp. 565–582.
[29] Dsrc: dedicated short range communication, http://www.leearmstrong.com/dsrc/dsrchomeset.htm.
[30] I.A. Sumra, H.B. Hasbullah, J.-l.B. AbManan, Attacks on security goals (confidentiality, integrity, availability) in vanet: a survey, in: Vehicular Ad-hoc Networks for Smart Cities, Springer, 2015, pp. 51–61.
[31] S. Saroiu, A. Wolman, Enabling new mobile applications with location proofs, in: Proceedings of the 10th Workshop on Mobile Computing Systems and Applications, ACM, 2009, p. 3.
[32] Y.S. Rao, R. Dutta, Computationally efficient secure access control for vehicular ad hoc networks, in: Information Systems Security, Springer, 2012, pp. 294–309.
[33] A.A.M. Green, M. Rushanan, libfenc: the functional encryption library, http://code.google.com/p/libfenc/.
[34] B. Lynn, Stanford pairings-based crypto library, http://crypto.stanford.edu/pbc/.
[35] B.W. John Bethencourt, Amit Sahai, Ciphertext-policy attribute-based encryption, http://hms.isi.jhu.edu/acsc/.
[36] C. Cseh, Architecture of the dedicated short-range communications (dsrc) protocol, in: Vehicular Technology Conference, vol. 3, IEEE, 1998, pp. 2095–2099.