天主教輔仁大學圖書資訊學系碩士班碩士論文

指導老師：李正吉 博士

車載通訊批次驗證及會議金鑰協議機制之研究

The Study of Batch Verification and Session Key Agreement Schemes for Vehicular Communications

研究生：賴彥銘 撰

中華民國 102 年 7 月

中文摘要

隨著無線網路技術的日新月異，其應用也日益廣泛。車載網路(Vehicular Ad-hoc Network，以下簡稱 VANET)，為一種將分散式網路拓樸 Ad-hoc 網路模型應用到車輛通訊的架構。VANET 可分為車輛對車輛的 V2V (Vehicular to Vehicular)與車輛對道路單位的 V2R (Vehicular to Roadside unit)。V2V 可讓車輛間建立簡單的通訊網路，使駕駛們在可交換並討論訊息。V2R 則可讓車輛對公共設施回報即時資訊，也可讓車輛可透過RSU連結到Internet，進行網路資料查詢等動作。另外，藉由掌握 RSU 所掌握之回報資訊與回報流量等，可輔助智慧型交通控制系統，達到交通系統效能最佳化。

隨著 VANET 應用範圍的提升，其安全性逐漸受到重視。除此之外，資訊的即時性也是VANET所面臨的挑戰。因此，本研究將著重VANET通訊協議的安全性與效率性，提出相關論述。我們將針對 V2R 的資訊認證與 V2V 的通訊金鑰建立加入以雙線性為基礎的批次執行技術，以提升系統效率並維持安全性。根據安全性與效率分析說明，我們所提出的機制相較於過去的方法顯得更加安全及更有效率。

關鍵字：車載網路、金鑰協定、資訊認證、批次執行、雙線性

ABSTRACT

Vehicular Ad-Hoc Network (VANET) is an application of Ad-Hoc Network, which can significantly improve the efficiency of transportation systems and allows the driver can exchange information via a privacy channel. The security is an important issues in the VANET system, because its significant impact, and the transportation systems may be paralyzed as a result of receiving the wrong traffic information. However, most of currently known schemes focus on a one by one basis. In real situation, the large amount of traffic flow will generate a lot of information at the same time. If the method is doing one by one, it is bound to lead to information delays, and the system will have difficulty to achieve real-time performance. Therefore, we shall propose two improved schemes based on the batch operation and bilinear pairing to make VANET more secure, efficient, and more suitable for practical use.

Keyword: batch operation, bilinear pairing, information verification, key agreement, VANET.

誌謝 (ACKNOWLEDGEMENTS)

一件事情的成功，其背後往往有許多來自許多貴人的支援。我從大學時，就在輔仁大學圖書資訊學系，連同研究所至今已經六年。在這六年當中，我吸收系上課程所提供的資訊成為自己的知識、利用系上所提供的各種機會茁壯自己、把握各位師長們所提供的機會磨練自己，最終的有形成果之一便是這本碩士論文。這當中遇到許多貴人，在他們的幫助下，我才能有今天的成果。

我首先要感謝的，是我的指導老師，李正吉老師。李正吉老師在我大學四年級時來到輔大圖資。老師對學生的熱情，大家有目共睹。而他的指導方式，嚴厲不失溫柔，積極不失耐心，尤其是他願意花費許多額外的時間來幫助我解決問題、校訂論文、潤稿等等，並提供許多他過去參與研究的心得、經驗讓我得以學習與成長。如果沒有李正吉老師的幫助，我不可能在如此短暫的時間便能展現成果，李正吉老師的付出，讓我銘感五內，謝謝李正吉老師。我也要感謝我的口試老師：謝建成老師與李俊達老師，他們所給予的精闢見解與延伸意見是我這本論文的重要基石，讓我的論文可以趨近完整，並且對於未來的發展方向有更開拓的視野，謝謝謝建成老師，謝謝李俊達老師。再來，我想感謝林麗娟老師、陳舜德老師與吳政叡老師。這三位老師在我大學生活中，肯定我的潛力，並提供了許多機會磨練我，讓我對自己越來越有自信，也無形之中培養了足以解決問題的實力與學習的興趣，我想對這三位老師致上由衷的感謝。

我也要感謝輔大圖資系這個大家庭。這個大家庭所提供的溫暖氛圍，讓我可以悠遊學習，而不會感到束縛受限。平易近人的老師，讓我勇於向老師請教問題，獲得更多額外的知識；友善的助教，協助我快速處理非課業的業務，讓我不用為瑣事煩心；充足的設備，讓我可以快速取得各種資訊，而不至於延誤研究進度。在這種環境下，我可以專心學習、研究，十分感謝系上所提供的一切。

我最想感謝的，當然是我的父母。我的父母在學習上一直支持我，讓我可以無後顧之憂地向前邁進。他們給我最大的自由，但不忘在我受挫時關心我，讓我可以前進時勇往直前，後退時又可安心地準備重新出發。謝謝父親與母親所提供的支援與支持，感謝你們。

最後，我還想感謝許多人，提供我機會、經驗、知識的學長姐；和我一起討論問題、解決問題的同儕；一起吃喝玩樂，盡情放鬆，然後再度往前的朋友，謝謝你們。這些幫助將成為我的養分，並在日後發揮它的價值。一段旅途的結束，象徵另一段旅途的開始。我在這個旅途的終點，感謝旅途上所有幫助過我的人們，並熱切期盼下一段旅途的開始。

謝謝大家

TABLE OF CONTENT

中文摘要 …… i

ABSTRACT …… ii

誌謝 …… iii

TABLE OF CONTENT …… v

LIST OF TABLES …… vii

LIST OF FIGURES …… viii

Chapter 1 Introduction …… 1

1.1 Research Motivation …… 1

1.2 Research Subjects …… 5

1.3 Thesis Organization …… 6

Chapter 2 Preliminaries …… 7

2.1 Bilinear Maps …… 7

2.2 Chinese Remainder Theorem …… 8

2.3 Homomorphism Encryption …… 9

Chapter 3 Toward A Secure Batch Verification with Group Testing for VANET · 10

3.1 Introduction …… 10

3.2 Background …… 12

3.3 Review of Zhang et al.'s Scheme ····· 14

3.4 Cryptanalysis of Zhang et al.'s Scheme ····· 18

3.5 The Batch Verification Scheme ····· 20

3.6 Analysis of the Batch Verification Scheme ····· 24

Chapter 4 An Efficient Multiple Establishing Session Key Scheme for Integrating Different Groups in VANET ····· 28

4.1 Introduction ····· 28

4.2 Review of the Yeh et al.'s Scheme ····· 30

4.3 Analysis of Yeh et al.'s Scheme ····· 34

4.4 The Multi Key Agreement Scheme ····· 36

4.5 Analysis of the Multi Key Agreement Scheme ····· 41

Chapter 5 Conclusions ····· 53

References ····· 54

LIST OF TABLES

Table 3.3.1 Notation of the batch verification 15

Table 3.6.1 Security comparison 25

Table 3.6.2 Comparison of three schemes in term of the computational complexity 27

Table 4.2.1 Notations of the Yeh et al.'s scheme 31

Table 4.4.1 Notations of the multi key agreement 37

Table 4.5.1 Comparison of transmission times 52

LIST OF FIGURES

Figure 1.1.1 VANET sketch 2

Figure 3.2.1 VANET network sketch 12

Figure 3.6.1 Effect on the batch verification delay 27

Figure 4.3.1 Message transference in traditional ElGamal encryption 35

Figure 4.3.2 Message transference in Yeh et al.'s scheme 35

Figure 4.5.1 Message transference of Yeh et al.'s scheme in groups combined 52

Figure 4.5.2 Message transference of the multi key agreement scheme in groups combined 52

Chapter 1 Introduction

1.1 Research Motivation

1.1.1 Background of VANET

Due to the development of wireless communication technologies, they have been used widely and have attracted great attention in recent years. Ad-Hoc network is a representative wireless's application. Ad-Hoc has some advantages, such as having fewer infrastructures, arranging a LAN (Local Area Network) quickly, and allowing its members to either join or leave easily. Because of these reasons, Ad-Hoc has become the first choice network model to use in order to establish a real-time LAN. This network model is suitable for an environment that changes frequently or that does not have enough of an infrastructure, i.e. a disaster area or a transportation system [9, 19, 36].

Vehicular Ad-Hoc Network (VANET) is the application of Ad-Hoc network with respect to vehicle communication. Each vehicle can use a device, called On-Board Units (OBUs), to communicate with each other vehicle, the Road Side Unit (RSU) or other infrastructures [8, 12, 37]. There are two types VANET: Vehicle-to-Vehicle (V2V) communication and Vehicle to RSU (V2R) communication [4, 8, 12, 37, 29, 30, 39, 41, 42]. Due to V2V, people can obtain more information and use the information to achieve road safety, such as maintaining a distance from other vehicles and rear vehicles. Furthermore, a group can establish simple communication networks and allow members to communicate with others. People can also communicate with RSU by V2R to download files from the Internet or ask the closest location information, such as the closest gas station and restaurant. In addition, users can query RSU about the local

situation in order to avoid traffic jams. Because RSU is an infrastructure, it can be an Internet node. Hence, people can use Internet services to upload or download files through RSU. On the other hand, traffic management is easy to carry out by combining the traffic system and the VANET system. Because RSU can collect and monitor traffic flow information, the traffic system can predict the traffic flow and control traffic signals to regulate the flow in real time. If necessary, traffic system can cooperate with the public affair vehicles, such as ambulances or fire engines, to improve the efficiency of solving any urgent task. Figure 1.1.1 is the sketch of VANET. On the right of figure 1.1.1, a lot of vehicles are waiting for the traffic light and they will report the waiting signals to the nearest RSU. When the RSU collects enough signals, it lets the traffic light be turned into green and the waiting vehicle can continue advancing. On the left of the picture, there is a traffic accident. The vehicle in the accident is telling police the accident place through RSU. In addition, a vehicle on the picture lower side is communicating with other vehicle by using V2V.

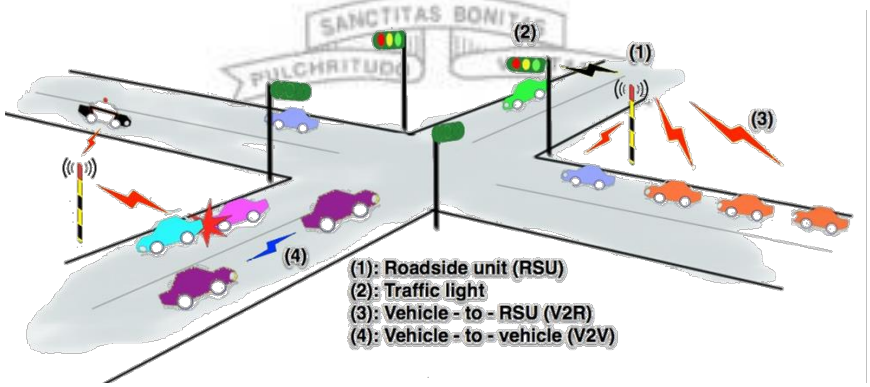


Figure 1.1.1 VANET sketch

Since VANET can provide people with many applications about traffic experience, the security issues in VANET are particularly important. The applications of VANET are in general grouped into two categories: public and privacy applications [15, 22, 31, 41].

1.1.2 Public Application

The public application means the transferred information are usually related to local information and traffic information, such as gas station query or traffic jam report and it's. Because that the public application always involves no-privacy information, it only need less encrypted. For this reason, it can be transferred faster than the privacy application. However, it still has some security challenges in operation. One of that is avoiding wrong messages, such as falsified messages, replayed messages, or malicious messages. The wrong messages maybe cause some poor situations such as the following.

(1) Wrong traffic flow messages:

The wrong traffic flow message may result in the traffic management system making wrong decisions. The wrong decision will cause the traffic lights of the heavy side to stay red and the other side to stay green.

(2) Wrong traffic stat messages:

The wrong traffic stat message may mislead driver into a traffic jam, and the traffic will be heavier.

(3) Wrong vehicles messages:

The wrong vehicles message may make the driver misread the safe distance, and crash into other vehicles.

(4) Falsified messages:

If an adversary falsifies a public affair vehicle signal, such as an ambulance's signal, he/she may compel the traffic light to cooperate with him/her and harm the driving right of other drivers.

Because VANET improves the traffic experience substantially, any secure leak of VANET may cause inestimable harms to the traffic system. For this reason, designing a secure scheme to ensure the confidence of VANET is the most important in VANET.

1.1.3 Privacy Application

Relative to the public application, the transferred information in the privacy application is always used for V2V's privacy communication, such as the group's goal, information about the vehicle, or other privacy information and needs complete encrypted to protect the privacy [22, 31, 41]. However, VANET is a wireless network environment, and messages traveling through such wireless network systems can be easily intercepted. V2V communications often involve private information exchange and therefore demand the establishment of secret session channels. Although the asymmetric cryptosystem is the more safe than symmetric cryptosystem, it is not suitable for VANET applications. One major reason is that VANET has only minimum infrastructure and may probably fail to provide the public key for the users when necessary [43]. Comparatively, the symmetric cryptosystem based on session keys is more suitable, and yet it also has an operational problem. If the session keys are fixed, then the vehicle has to pre-obtain all the session keys of the communication targets; that is to say, the vehicle has to store a large number of session keys in advance, which is both insecure and inefficient [1, 19, 20, 27]. Comparatively, the symmetric cryptosystem based on session keys is more suitable,

and yet it also has an operational problem. If the session keys are fixed, then the vehicle has to pre-obtain all the session keys of the communication targets; that is to say, the vehicle has to store a lot of session keys in advance, which is both insecure and inefficient

1.1.4 Batch Operation

To ensure the information that is instant is another challenge in the VANET environment. Because that the members in the VANET environment are always on the move, the outmoded information is without any worth [4, 23, 29]. Supposing that a driver wants, to inquire the next Interchange to leave from expressway, the outmoded information will make him miss the opportunity and he must spend more time getting back to the correct way. However, the traffic flow has its periodicity, for example, the flow is high at on and off duty, and low at ordinary times. When the flow is high, the traditional one by one verification mechanism will let the system is overload with too many requests. In this case, some requests may be delayed or dropped [41]. Batch operation is a way to solve this problem. By reducing the most complex operations, the operations costs can avoid the linear growth or exponent growth. For this reason, batch operation can help the researchers to design a more suitable protocol for VANET applications.

1.2 Research Subjects

In this study, we focus on the batch operation in VANET applications. There are two subjects in this thesis. The first is batch verification for public V2R environment. In 2011, Zhang et al. proposed a new scheme for VANET batch verification [41]. Their scheme is based on bilinear pairing and use addition operations to batch verify multiple signatures simultaneously. As an addition operation is simpler than any exponent operations, the

Zhang et al.’s scheme is more efficient. However, Zhang et al.'s scheme has some weaknesses. Therefore, how to improve the Zhang et al.'s scheme is the first goal of this thesis.

The second subject is batch session keys established for privacy V2V communication. Although a lot of useful achievements for key agreement were proposed in the past, they are almost suitable to one by one session key established. In 2012, Yeh et al. proposed a novel framework for batch authenticated and session keys established [40]. The Yeh et al.'s scheme allows a new member establishes multi session keys with different members simultaneously when joining to an existent group. However, the Yeh et al.'s scheme is not suitable to the VANET environment. Because that the groups or teams in VANET environment are usually temporary and changeable, it may be happened that two or more teams are combined and the members need to establish new session key with the new partners came from different groups originally. In this case, the Yeh et al.'s is limited to a certain extent. For this reason, we will extend the application of the Yeh et al.'s scheme, and the new scheme is more suitable to establish the batch session keys for privacy V2V.

1.3 Thesis Organization

The remainder of this thesis is organized as follows. In Chapter 2, we introduce the preliminaries that are used in this thesis briefly. Then, we describe the Zhang et al.'s scheme for public VANET batch authentication [41] and propose an improved scheme in Chapter 3. In Chapter 4, we review the Yeh et al.'s scheme [40] and propose an extended scheme that is suitable for the VANET's many-to-many situation. Finally, our conclusions are shown in Chapter 5.

Chapter 2 Preliminaries

For designing suitable schemes for VANET, we use some mathematical tools, which involve bilinear maps and Chinese remainder theorem. In addition, we also use the homomorphism encryption to help the deigned scheme to become more secret. In this section, we introduce those tools as follows briefly.

2.1 Bilinear Maps

Bilinear maps is a power tool that is good at batch operation, and we briefly introduce the bilinear maps as follows [20, 26].

(1) Define a bilinear map ê:

i. Let G be a cyclic additive group, and G_T be a cyclic multiplicative group generated by P. G and G_T have the same prime order q, and $|G|=|G_T|$.

ii. Define $\hat{e}: G\times G\rightarrow G_T$ be a bilinear map.

(2) Bilinear map has the following properties:

i. Bilinear:

For the all $P, Q, R\in G$, $\hat{e}(Q, P+R) = \hat{e}(P+R, Q) = \hat{e}(Q, P)\cdot \hat{e}(Q, R)$.

For the all $P, Q\in G$, and $a, b \in Z_q^*$, $\hat{e}(aQ, bP) = \hat{e}(bQ, aP) = \hat{e}(Q, P)^{ab}$.

ii. Non-degenerate: There exist $P, Q \in G$ such that $\hat{e}(P, Q) \neq 1_{G_T}$, where 1_{G_T} is the identity element of G_T.

iii. Computable: There is an efficient algorithm to compute $\hat{e}(P, Q)$ for each $P, Q \in G$.

Bilinear maps can be constructed by utilizing modified elliptic curves [12, 24, 33]. They share the same characteristic with elliptic curves: Given $P, Q \in G$ and $a \in Z_q^*$, $Q=aP$, and $\{P, Q\}$ are known. To derive the integer a from Q and P is to solve the Elliptic Curve Discrete Logarithm Problem (ECDLP).

2.2 Chinese Remainder Theorem

The Chinese Remainder Theorem (CRT) is an ancient math problem published by a famous Chinese military general, strategist, philosopher, and mathematician Sun Tzu in around 3rd to 5th century. By CRT, we can hide a lot of different secret values in a large number X and broadcast X to different receivers [17, 40]. Each receiver can decrypt the value X and obtain correct secret value respectively. We use an example to exam CRT. Let there are a sender A and n receivers: $R_1, R_2 \dots R_n$. The sender A has agreed the session key $SK_1, SK_2 \dots SK_n$ with each receiver. If A wants to send messages $M_1, M_2 \dots M_n$ to receivers $R_1, R_2 \dots R_n$ individually, A can encrypt those messages in a assemble value X and send the X to those receivers. The value X has the following characteristic:

$$X \equiv M_1 \bmod SK_1, X \equiv M_2 \bmod SK_2, \dots X \equiv M_i \bmod SK_i, \dots X \equiv M_n \bmod SK_n$$

Due to this characteristic, each receiver R_i can obtain the correct value M_i, where i=1, 2…n. The assembly value is generated as follows.

$$L = \prod_{i=1}^{n} SK_i$$

$$A_i \times \left(\frac{L}{SK_i}\right) \equiv 1 \bmod SK_i$$

$$X \equiv \sum_{i=1}^{n} \left(\frac{L}{SK_i} \times M_i \times A_i\right) \bmod L$$

2.3 Homomorphism Encryption

Homomorphism encryption is a good tool to transfer aggregate information from multi sources to a destination [33]. It enables different users to encrypt different information by using the same key, which is owned by the target user, and the target user can then obtain an aggregation after decrypting the ciphertext. Quite a number of homomorphism encryption methods have been proposed based on different cryptosystems, and many different types of aggregate information can be processed, such as addition, multiplication, XOR, etc. In this thesis, we use the Paillier cryptosystem [28] to help us design the proposed scheme.

Here is an example that shows in detail how the Paillier cryptosystem works [28]:

(1) Let $N = p \cdot q$ and $\lambda = lcm(p-1, q-1)$, where $\{p, q\}$ are two primes and lcm means the least common multiple .

(2) Select two random numbers $g_1 \in Z^*_{N^2}$ and $g_2 \in Z^*_N$.

(3) Define encrypt function $E(M) = g_1^M g_2^N \bmod N^2$, where M is the plaintext.

(4) Define decrypt function $D(c) = \frac{L(c^\lambda \bmod N^2)}{L(g_1^\lambda \bmod N^2)} \bmod N$, where c is the ciphertext and $L(.)$ is defined such that $L(u) = \frac{u-1}{N}$ for each $u < N^2$ and $u \equiv 1 \bmod N$.

The characteristic of Paillier cryptosystems is as follows.

(5) $D(E(M_1) \cdot E(M_2) \bmod N^2) = (M_1 + M_2) \bmod N$

Due to this characteristic, multi sources can submit information to a target user in an aggregation.

Chapter 3 Toward A Secure Batch Verification with Group Testing for VANET

3.1 Introduction

Because VANET improves the traffic experience substantially, any secure leak of VANET may cause inestimable harms to the traffic system. To ensure both the integrity of the messages and non-repudiation is indispensable. A simple solution is to sign each message with a digital signature before the message is sent. In 1976, Diffie and Hellman proposed an idea about public-key cryptography [10]. Two years later, Rivest et al. proposed a novel scheme to accomplish Diffie-Hellman's idea, called RSA algorithm [32]. In 2007, Raya and Hubaux [30] proposed appropriate security architecture for VANET. There is a PKI (Public Key Infrastructure) certificate issues in their scheme. The RSU and the OBU can mutually authenticate by means of the other's public key and establish a session key for communication. However, most of the traditional signature schemes verify the received signatures one by one. When the traffic is heavy, the verifier will receive a lot of signatures. Verifying a large number of signatures sequentially will to rate a long time, and the information with the signature will be delayed. Because the traffic situations are always changed, instantaneity is a very important issue for the traffic information [4, 23, 29]. If the information is not fresh, it cannot explain the real traffic situation and help people or traffic management system make decisions, and the information will lose its value [8, 37, 41, 42]. To solve the verification bottleneck problem, a lot of related schemes have been proposed. In 1990, Fiat proposed the first batch cryptography scheme based on RSA [11]. In 2007, Lin et al. proposed a group signature scheme based on bilinear pairing to improve the authentication efficiency [23]. Because

the verifier can verify multiple signatures simultaneously in Lin et al.'s scheme, the cost of computation time will not grow linearly with the amount of the signature. Unfortunately, Lin et al.'s scheme uses a lot of exponent operations, and it has complex computing process. In 2011, Zhang et al. [41] and Huang et al. [14] proposed a new scheme respectively. Both of their schemes are based on bilinear pairing and use addition operations to batch verify multiple signatures simultaneously. As an addition operation is simpler than any exponent operations, both of the two schemes are more efficient. Because batch verifying is more efficient than single verifying when the verifier has to verify a large number of signatures. However, Zhang et al.'s scheme has some weaknesses. First, Zhang et al.'s scheme is vulnerable to the replaying-attack. Because of this weakness, an adversary can simulate a fake situation, such as a traffic jam, by collecting the vehicle messages and signatures in the corresponding situation and replaying them. Second, Zhang et al.'s scheme doesn't achieve the signature non-repudiation. A malicious driver can broadcast wrong information to mislead other drivers and repudiate the behavior when the traffic manager traces him/her by his/her signature. In Huang et al.'s scheme, which is known as ABAKA, the scheme also doesn't achieve the signature non-repudiation. Wanga and Zhang pointed out this weakness in 2012 [38]. Hence, ABAKA is not suitable for VANET. The details of ABAKA can refer to [14]. For this reason, we want to propose an improved scheme to enhance the security and keep the efficiency of Zhang et al.'s scheme. The improved scheme can make the VANET information verification be more suitable.

In this section, we will describe the weaknesses of Zhang et al.'s scheme, and propose an improved scheme. The section is organized as follows. In subsection 3.2, we present the background, which includes the network model and equipment and security

requirements. After that, we describe Zhang et al.'s scheme in subsection 3.3 and provide our analysis in subsection 3.4. In subsection 3.5, we will propose an improved scheme and present an analysis of the batch verification scheme in subsection 3.6 finally.

3.2 Background

3.2.1 Network Model and Equipment

Two-layer network model is often used in vehicular network [2, 14, 41]. As its name suggests, there are two layers in the network model: top layer and lower layer. The sketch is shown as figure 3.2.1.

The top layer is comprised of a Trust Authority (TA) and application servers. We assume that TA can be completely trusted, and it is responsible for pre-assigning secure information for each vehicle. In general, TA is ordinarily off-line with other vehicles, and responsible for tracing the real identity of vehicles in case that disputes happens. The application servers for public applications, such as traffic management center, communicate with RSUs and provide services or information. In the lower layer, vehicles and RSUs can communicate with others based on the dedicated short-range communications (DSRC) protocol [2]. Each vehicle has its own public and private key-pairs for signing each message before the message is sent. Messages and signatures will sent to the sender's neighboring RSU, and the RSU will verify the digital signatures after receiving those information. Each vehicle has to be equipped with a tamper-proof device, which is a secure storage for secrets. We assume that the tamper-proof device is always credible and its information is never been disclosed. The device will pre-load some secure values, such as real identity of vehicle and secret key of system. The computing process of vehicle is also included in this device and the value is never disclosed.

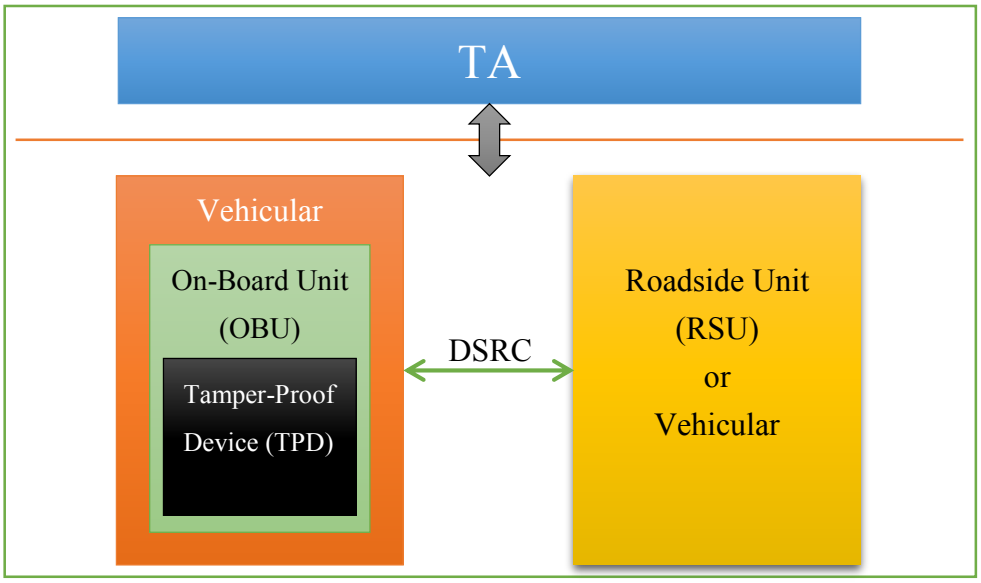


Figure 3.2.1 VANET network sketch

3.2.2 Security Requirements

To protect the privacy of the users, communication security is important. In VANET communication, security issues are also very important. In this field, we can generalize three security requirements as follows [7, 42].

(1) Message authentication:

Ensuring that a message was sent from a legitimate user and the integrity of message wasn't broken is a primary issue.

(2) User privacy preserving:

In VANET, communications are always transmitted via a wireless network. Compared to a wire network, wireless is easier intercepted, overheard, and traced. The system has to protect the privacy of a legitimate user, including the user's real identity or other individual information.

(3) Audit-ability:

To avoid the inside user using the user privacy preserving to broadcast malicious message which maybe mislead other legitimate user, systems should have a mechanism for retrieving the real identity of a malicious user.

3.3 Review of Zhang et al.'s Scheme

There are three subsections in Zhang et al.'s scheme [41], including key generation and pre-distribution, pseudo identity generation and message signing, and message verification. The notation is shown in table 3.3.1. We briefly describe them as follows.

3.3.1 Key Generation and Pre-Distribution

In Zhang et al.'s scheme, *TA* is responsible for setting up the system parameters for each vehicle and RSU as follows.

(1) Let G be a cyclic additive group generated by P, and G_T be a cyclic multiplicative group and G and G_T Have the same prime order q. After that, let $\hat{e}: G \times G \rightarrow G_T$ be a bilinear map.

(2) Choose two random numbers $\{s_1, s_2\} \in Z_q^*$ as its two master keys, and compute $P_{pub1} = s_1 P, P_{pub2} = s_2 P$ as its public keys. These two master keys $\{s_1, s_2\}$ of the *TA* are pre-loaded in each vehicle's tamper-proof device.

(3) The public parameters $\{G, G_T, Q, P, P_{pub1}, P_{pub2}\}$ are pre-loaded in each RSU and vehicle.

(4) Each vehicle is assigned its real identity, denoted as $RID \in G$, and password, denoted as *PWD*. Both *RID* and *PWD* are stored in the tamper-proof device.

Table 3.3.1 Notation of the batch verification

V_i	The *i*-th vehicle
RSU	A roadside unit
TA	A trust authority
TPD	A tamper-proof device
s_1, s_2	The private master key of the system
P_{pub1}, P_{pub2}	The public key of the *TA*
RID_i	The real identity of V_i $RID_i \in G$
PWD_i	A password of V_i
ID^i	A pseudo identity of the vehicle V_i, $ID^i = (ID_1^i, ID_2^i)$
SK^i	A private key of the vehicle V_i, $SK^i = (SK_1^i, SK_2^i)$
M_i	A message sent by the vehicle V_i
$h(), h_2()$	The one-way hash functions $h: \{0,1\}^* \rightarrow Z_q^*, h_2: \{0,1\}^* \rightarrow Z_q^*$
$H()$	A map to point hash function, $H: \{0, 1\}^* \rightarrow G$
$\|$	Message concatenation operation
T_i	A timestamp generated by V_i
Vec_i	A vector used to distinguish signatures, *i=1, 2, ..., n*

3.3.2 Pseudo Identity Generation and Message Signing

To achieve user anonymity, each vehicle has to generate a pseudonym before commutation. The details of this phase are shown as follows.

(1) The vehicle V_i inputs its unique real identity RID_i and the password PWD_i to initiate a pseudo identity generation process.

(2) After verifying RID_i and PWD_i, *TPD* chooses a random number *r* and computes pseudo $ID^i = \{ID_1^i, ID_2^i\}$ and SK^i=$\{SK_1^i, SK_2^i\}$.

ID_1^i=*rp*

ID_2^i= $RID_i \oplus H(rP_{pub1})$

SK_1^i=$s_1 ID_1^i$

SK_2^i=$s_2 h(ID_1^i \parallel ID_2^i)$

(3) After that, *TPD* outputs ID^i and SK^i, and V_i can sign messages by using those values.

(4) Each message M_i has to be signed before sent. V_i signs M_i as $\sigma_i = SK_1^i + h(M_i)\, SK_2^i$. Subsequently, V_i sends the final message $\{ID^i, M_i, \sigma_i\}$ to its neighboring *RSU*.

If V_i broadcasts a malicious message, *TA* can trace the RID_i of V_i by computing RID_i=$ID_2^i \oplus H(s_1 ID_1^i)$.Therefore, once a signature is in dispute, the *TA* has the trace ability to find the *RID* of vehicle from the disputed message.

3.3.3 Message Verification

The message verification process of Zhang et al.'s scheme has two versions: single message verification and batch message verification. We briefly describe them as follows.

3.3.4 Single Message Verification

When each RSU receives any final message, such as $\{ID^i, M_i, \sigma_i\}$ from a vehicle, it will verify the message's validity. If $\hat{e}(\sigma_i, P) = \hat{e}(ID_1^i, P_{pub1}) \cdot \hat{e}(h(M_i)H(ID_1^i \| ID_2^i), P_{pub2})$, the message is legal and unaltered. The proof is shown as follows.

$$\hat{e}(\sigma_i, P)$$

$$= \hat{e}\left(SK_1^i + h(M_i)SK_2^i, P\right)$$

$$= \hat{e}\left(SK_1^i, P\right) \cdot \hat{e}\left(h(M_i)SK_2^i, P\right)$$

$$= \hat{e}(s_1 ID_1^i, P) \cdot \hat{e}(h(M_i)s_2 H(ID_1^i \| ID_2^i), P)$$

$$= \hat{e}(ID_1^i, s_1 P) \cdot \hat{e}(h(M_i)H(ID_1^i \| ID_2^i), s_2 P)$$

$$= \hat{e}(ID_1^i, P_{pub1}) \cdot \hat{e}(h(M_i)H(ID_i^1 \| ID_2^i), P_{pub2})$$

3.3.5 Batch Message Verification

If a RSU receives a number of large messages, denoted as $\{ID^1, M_1, \sigma_1\}$, $\{ID^2, M_2, \sigma_2\}$, $\{ID^3, M_3, \sigma_3\}$... $\{ID^n, M_n, \sigma_n\}$, in a short span, the RSU can verify the messages' validity simultaneously by means of batch message verification. If

$$\hat{e}\left(\sum_{i=1}^{n} \sigma_i, P\right) = \hat{e}\left(\sum_{i=1}^{n} ID_1^i, P_{pub1}\right) \cdot \hat{e}\left(\sum_{i=1}^{n} (h(M_i)H(ID_1^i \| ID_2^i), P_{pub2}\right)$$

, the batch of messages is legal and unaltered. The proof of this equation is as follows:

$$\hat{e}\left(\sum_{i=1}^{n} \sigma_i, P\right)$$

$$= \hat{e}\left(\sum_{i=1}^{n} (SK_i^1 + h(M_i)SK_i^2), P\right)$$

$$= \hat{e}\left(\sum_{i=1}^{n}(SK_i^1), P\right) \cdot \hat{e}\left(\sum_{i=1}^{n}(h(M_i)SK_i^2), P\right)$$

$$= \hat{e}\left(\sum_{i=1}^{n}(SK_i^1), P\right) \cdot \hat{e}\left(\sum_{i=1}^{n}(h(M_i)s_2H(ID_1^i \| ID_2^i), P\right)$$

$$= \hat{e}\left(\sum_{i=1}^{n}s_1ID_1^i, P\right) \cdot \hat{e}\left(\sum_{i=1}^{n}(h(M_i)H(ID_1^i \| ID_2^i), s_2P\right)$$

$$= \hat{e}\left(\sum_{i=1}^{n}ID_1^i, s_1P\right) \cdot \hat{e}\left(\sum_{i=1}^{n}(h(M_i)H(ID_1^i \| ID_2^i), P_{pub2}\right)$$

$$= \hat{e}\left(\sum_{i=1}^{n}ID_1^i, P_{pub1}\right) \cdot \hat{e}\left(\sum_{i=1}^{n}(h(M_i)H(ID_1^i \| ID_2^i), P_{pub2}\right)$$

3.4 Cryptanalysis of Zhang et al.'s Scheme

Zhang et al. proposed an efficient batch message verification to solve the verification bottleneck problem. However, the Zhang et al. scheme has two weaknesses, i.e. it is vulnerable to the replaying attack and to failure to achieving non-repudiation. The details of the two weaknesses of Zhang et al.'s scheme are shown as follows.

3.4.1 Replaying Attack

Zhang et al.'s scheme is vulnerable to the replaying attack. We assume an adversary can intercept a public affair vehicle message and signature. He/she can replay the information to mislead the traffic management system when he/she needs. On the other situation, an adversary can intercept a lot of signatures from different vehicles when those

vehicles are in a traffic jam, and replay those signatures to invent a fake traffic jam and mislead other vehicles in order to avoid the jammed sections.

3.4.2 Not Achieving Non-Repudiation

Zhang et al.'s batch message verification is very efficient. However, the batch verification scheme makes a leak, which allows the malicious user to deny his/her signatures. Assume a malicious user generates several different messages and signatures, such as $\{ID^1, M_1, \sigma_1\}$, $\{ID^2, M_2, \sigma_2\}$, $\{ID^3, M_3, \sigma_3\}$, and swaps their contents to becomes $\{ID^1, M_1, \sigma_3\}$, $\{ID^2, M_2, \sigma_1\}$, $\{ID^3, M_3, \sigma_2\}$. After that, the malicious user sent those changed messages and signatures to its neighboring RSU. If the RSU uses a batch message verification process to verify those signatures, it will consider that those changed messages and signatures are legal. The proof is shown as follows.

$$\hat{e}\left(\sum_{i=1}^{3}\sigma_i, P\right)$$

$$= \hat{e}(\sigma_1 + \sigma_2 + \sigma_3, P)$$

$$= \hat{e}(\sigma_3 + \sigma_1 + \sigma_2, P)$$

$$= \hat{e}\left(\sum_{i=1}^{3} ID_1^i, P_{pub1}\right) \cdot \hat{e}\left(\sum_{i=1}^{3} h(M_i)H(ID_1^i \parallel ID_2^i), P_{pub2}\right)$$

Although the orders of those signatures have been changed, their sum is not changed. However, those messages and signatures aren't conformed obviously, there signatures can't pass if the RSU uses single message verification process to verify them one by one. For this reason, the malicious user can deny his/her signatures.

3.5 The Batch Verification Scheme

To overcome those weaknesses of Zhang et al.'s scheme, we propose an improved batch verification scheme. In the batch verification scheme, we extend the framework of Zhang et al.'s scheme. In the batch verification scheme, we also use a two-layer vehicular network model, and we require each vehicle to have a tamper-proof device. The notation of the batch verification scheme is also shown in table 3.3.1.

The batch verification scheme also includes key generation and pre-distribution, pseudo identity generation and message signing, and message verification. The differences between Zhang et al.'s scheme and the batch verification scheme are pseudo identity generation and message signing, and message verification. We explain them as follows.

3.5.1 Key Generation and Pre-Distribution

In the batch verification scheme, *TA* is also responsible for setting up the system parameters for each vehicle and RSU. The process of this phase is the same as Zhang et al.'s scheme, and the difference is easily discerned in the subsequent subsections.

3.5.2 Pseudo Identity Generation and Message Signing

To achieve user anonymity, each vehicle has to generate a pseudonym before commutation. In this subsection, we add a timestamp T_i to overcome the replaying attack and use a one-way hash function $h_2()$ instead of the map to point function $H()$. The details of this phase are shown as follows.

(1) The vehicle V_i inputs its unique real identity RID_i and the password PWD_i to initiate pseudo identity generation process.

(2) After verifying RID_i and PWD_i, *TPD* chooses a random number *r*, sets a current timestamp T_i, and computes pseudo ID^i=$\{ID_1^i, ID_2^i\}$ and SK^i=$\{SK_1^i, SK_2^i\}$.

$ID_1^i = rP$

$ID_2^i = RID_i \oplus H(rP_{pub1})$

$SK_1^i = s_1 ID_1^i$

$SK_2^i = s_2 h_2(ID_1^i \parallel ID_2^i \parallel T_i)P$

(3) After that, *TPD* outputs ID^i and SK^i, and V_i can sign messages using ID^i and SK^i.

(4) Each message M_i has to be signed before sent. V_i signs M_i as $\sigma_i = SK_1^i + h(M_i)SK_2^i$. Subsequently, V_i sends the final message $\{ID^i, M_i, \sigma_i, T_i\}$ to its neighboring *RSU*.

If V_i broadcasts a malicious message, *TA* can trace the RID_i of V_i by computing RID_i=$ID_2^i \oplus H(s_1 ID_1^i)$.Therefore, once a signature is in dispute, the *TA* has the trace ability to find the *RID* of vehicle from the disputed message.

3.5.3 Message Verification

When each RSU receives any final message, such as $\{ID^i, M_i, \sigma_i, T_i\}$ from a vehicle, it will check the message's T_i. If $T_r - T_i < T_\Delta$, where T_r is the received-time of the message and T_Δ is the predefined endurable transmission delay, RSU either continues the verification process, or else rejects the final message. The message verification process of the batch verification scheme also has two versions: single message

verification and batch message verification. The details of these two versions are described as follows.

3.5.4 Single Message Verification

If the RSU just receives a few final messages in a span, it can verify the message's validity one by one. For each signature, if $\hat{e}(\sigma_i, P) = \hat{e}(ID_1^i, P_{pub1}) \cdot \hat{e}(h(M_i)h_2(ID_1^i \| ID_2^i \| T_i)P, P_{pub2})$, the message is legal and unaltered. Our proof is as follows.

$$\hat{e}(\sigma_i, P)$$
$$= \hat{e}(SK_1^i + h(M_i)SK_2^i, P)$$
$$= \hat{e}(SK_1^i, P) \cdot \hat{e}(h(M_i)SK_2^i, P)$$
$$= \hat{e}(s_1 ID_1^i, P) \cdot \hat{e}(h(M_i)s_2 h_2(ID_1^i \| ID_2^i \| T_i)P, P)$$
$$= \hat{e}(ID_1^i, s_1 P) \cdot \hat{e}(h(M_i)h_2(ID_1^i \| ID_2^i \| T_i)P, s_2 P)$$
$$= \hat{e}(ID_1^i, P_{pub1}) \cdot \hat{e}(h(M_i)h_2(ID_1^i \| ID_2^i \| T_i)P, P_{pub2})$$

3.5.5 Batch Message Verification

If an RSU receives a number of large messages, denoted as $\{ID^1, M_1, \sigma_1\}$, $\{ID^2, M_2, \sigma_2\}$, $\{ID^3, M_3, \sigma_3\}$... $\{ID^n, M_n, \sigma_n\}$, within a short span, the RSU can verify the messages' validity simultaneously by batch message verification. In this subsection, we add a vector parameter Vec_i to overcome the weakness of Zhang et al.'s scheme. Before batch message verification, RSU distributes Vec_i to each message and signature, where Vec_i's value are a random number and ranges between 1 and *x*, where *x* is a small value and doesn't make the overhead of computation. After that, RSU starts the batch message verification. If

$$\hat{e}\left(\sum_{i=1}^{n} Vec_i\sigma_i, P\right)$$

$$= \hat{e}\left(\sum_{i=1}^{n} Vec_i ID_1^i, P_{pub1}\right) \cdot \hat{e}\left(\left(\sum_{i=1}^{n} Vec_i h(M_i) h_2\left(ID_1^i \parallel ID_2^i \parallel T_i\right)\right)P, P_{pub2}\right)$$

, the batch of messages are legal and unaltered. The proof is as follows.

$$\hat{e}\left(\sum_{i=1}^{n} Vec_i\sigma_i, P\right)$$

$$= \hat{e}\left(\sum_{i=1}^{n} Vec_i\left(SK_1^i + h(M_i)SK_2^i\right), P\right)$$

$$= \hat{e}\left(\sum_{i=1}^{n} Vec_i SK_1^i, P\right) \cdot \hat{e}\left(\sum_{i=1}^{n} Vec_i\, h(M_i)SK_2^i, P\right)$$

$$= \hat{e}\left(\sum_{i=1}^{n} Vec_i\, s_1 ID_1^i, P\right) \cdot \hat{e}\left(\sum_{i=1}^{n} Vec_i\, h(M_i) s_2 h_2\left(ID_1^i \parallel ID_2^i \parallel T_i\right)P, P\right)$$

$$= \hat{e}\left(\sum_{i=1}^{n} Vec_i\, s_1 ID_1^i, P\right) \cdot \hat{e}\left(\left(\sum_{i=1}^{n} Vec_i\, h(M_i) h_2\left(ID_1^i \parallel ID_2^i \parallel T_i\right)\right)s_2 P, P\right)$$

$$= \hat{e}\left(\sum_{i=1}^{n} Vec_i\, ID_1^i, s_1 P\right) \cdot \hat{e}\left(\left(\sum_{i=1}^{n} Vec_i\, h(M_i) h_2\left(ID_1^i \parallel ID_2^i \parallel T_i\right)\right)P, s_2 P\right)$$

$$= \hat{e}\left(\sum_{i=1}^{n} Vec_i\, ID_1^i, P_{pub1}\right) \cdot \hat{e}\left(\left(\sum_{i=1}^{n} Vec_i\, h(M_i) h_2\left(ID_1^i \parallel ID_2^i \parallel T_i\right)\right)P, P_{pub2}\right)$$

3.6 Analysis of the Batch Verification Scheme

3.6.1 Security Analysis

In this subsection, we analyze the security of the proposed batch verification scheme in terms of the security requirements, which includes message authentication, user privacy preserving, audit-ability, as follows.

(1) Message authentication:

The message authentication is the most basic security requirement to ensure the legality of a message's source and the integrity of a message in any communication. In the batch verification scheme, σ_i not only uses a one-way hash function to pack the message M_i, but also uses a current timestamp T_i to generate SK_2^i in order to resist the replaying attack and ensure that the signature σ_i is fresh. The batch verification scheme also inherits the advantage of Zhang et al.'s scheme, include that it is difficult to derive the private keys SK_1^i and SK_2^i by way of ID^i, P_{pub1}, P_{pub2}, P, and $H(ID_1^i \| ID_2^i \| T_i)$ [41]. We not only overcame the replaying attack, but also proposed a solution to the other problem, non-repudiation. In the batch verification scheme, we used a vector parameter Vec_i to avoid user swap of the M_i and σ_i. If a malicious user wants to deny the signatures by swapping M_i and σ_i, his/her signatures will result in the batch message verification failing. Table 3.6.1 is a comparison between the batch verification scheme and other schemes which in the same field.

(2) User privacy preserving:

If an adversary attempts to use the information, which is intercepted from public communicating environment, to trace a specific user, he/she needs to determine the

relation between each communication. In the batch verification scheme, all of information sent by a user is changed in each communication. Therefore, a person's ID^i is converted by an unknown random number *r*. For this reason, we claim the batch verification scheme both achieves and preserves the user anonymity and user privacy.

(3) Audit-ability:

To avoid the user privacy preserving abused by the malicious behaviors, the malicious user should have *TA* traceability, where the traceability is also called conditional privacy [37]. In the batch verification scheme, the *TA* can trace the RID_i of V_i as the subsection 3.5.2 explains. When a user attempts to use malicious information to mislead others, the TA can trace the *RID* of the malicious user, and stop the right of the malicious user.

Table 3.6.1 Security comparison

	Batch message verification	Avoiding any replaying attack	Avoiding non-repudiation
Our scheme	✓	✓	✓
Zhang et al.'s scheme [41]	✓	×	×
ABAKA [14]	✓	✓	×

3.6.2 Performance Evaluation

We evaluate the performance of the batch verification scheme in this subsection. Verification delay is the most important issue, which maybe affects the value of

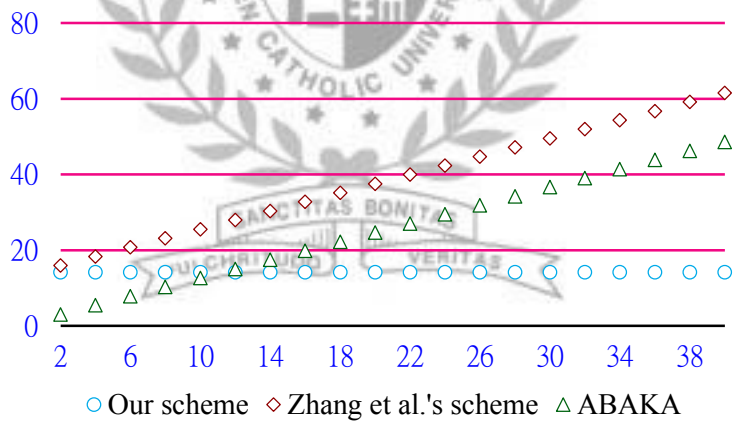
information. The different calculations in the batch verification scheme includes one point multiplication over an elliptic curve, notated T_{mul}, map to point hash operation, notated T_{mtp}, and pairing operation, and notated T_{par}. We adopts the MNT curve [14, 25, 41], which embeds degree k = 6 and 160-bit q, running on an Intel Pentium IV 3.0 GHZ machine. The following results are obtained: T_{mul} is 0.6 *ms*, T_{par} is 4.5 *ms*, and T_{mtp} is 0.6 *ms*. We compare the computational complexity of the batch verification scheme with Zhang et al.'s scheme and ABAKA in table 3.6.2. Although the batch verification scheme has to compute $Vec_i\sigma_i$, $Vec_i ID_1^i$, and $Vec_i h(M_i)h_2(ID_1^i \parallel ID_2^i \parallel T_i)$, the range of Vec_i is very small, such as 1 to 10, and the cost of Vec_i's computation is negligible. In fact, the real program design can use addition operation instead of multiplication operation, such as letting σ_i plus Vec_i times instead of computing $Vec_i\sigma_i$. On the other hand, we use a one-way hash function $h_2()$ instead of the map to point function $H()$ and reduce point multiplication over an elliptic curve to improve the performance. Hence, the efficiency of the batch verification scheme is more efficient than Zhang et al.'s scheme.

We use the results of the MNT curve and the value of performance comparison to forecast the effect on the batch verification delay of compared schemes in figure 3.6.1. We let *x*-axis mean the number of verifying signatures (*n*) and *y*-axis means the delay time (unit: *ms*). The figure 3.6.1 shows the situation while range of *n* is 1 to 40. We can find the slope of our scheme is the lowest. According to figure 3.6.1, although the effect of the batch verification scheme isn't the best when *n* is lower than 10, it is speeder than others when *n* become larger. When *n* is 100, the delay of ABAKS's batch verification is 120.6 *ms*, Zhang's is 133.5 *ms* and the batch verification scheme's is 14.1 *ms*. When *n* is 1000, the delay of ABAKS's batch verification is 1200.6 *ms*, Zhang et al.'s scheme is 1213.5 *ms* and the batch verification scheme's still maintains 14.1 *ms*; obviously, the batch

verification scheme is the most efficient. In addition, the batch verification scheme is more secure than ABAKA and Zhang et al.'s scheme. For this reason, Our scheme is the most suitable for VANET.

Table 3.6.2 Comparison of three schemes in term of the computational complexity

	Signal Verification	Batch Verification
Our scheme	$3T_{par} + T_{mul}$	$3T_{par} + T_{mul}$
Zhang et al.'s scheme [41]	$3T_{par}+T_{mtp}+T_{mul}$	$3T_{par}+nT_{mtp}+nT_{mul}$
ABAKA [14]	$3T_{mul}$	$(2n+1)T_{mul}$
*n: number of verifying signatures		



*x-axis: the number of verifying signatures (n)

*y-axis: the delay time (unit: *ms*)

Figure 3.6.1 Effect on the batch verification delay

Chapter 4 An Efficient Multiple Establishing Session Key Scheme for Integrating Different Groups in VANET

4.1 Introduction

In VANET environment, the driver can communicate with another driver and share or discuss some privacy information though V2V, which is a safety channel [8, 12, 37]. To build the safety channel, we can draw support from communication security field to reach it. In communication security field, the Diffie-Hellman key exchange algorithm was proposed to overcome this problem so as to maintain communication security in 1976 [10]. The method provides a way of real-time key generation, and by employing a key exchange protocol (also called key agreement protocol) to generate a session key for a user to start a communication session, the scheme saves the trouble of having users pre-store all the session keys of their communication targets. Besides, the forward secrecy can also be guaranteed even if one of the communication sessions is broken. The key agreement protocol generates one temporary session key at a time, and the temporary session key is only valid during the specific communication session it is created for and will expire after that. In case at a point of time many communication sessions are to be started, then the key agreement protocol will be extremely busy, and the communication system users will have to wait for their turn [18, 21]. If there are too many communication targets waited for agreeing the session in the same time, it has to spend a lot of time [40]. Assuming that a new user *R* want to join an exited group *A's group* that originally has twenty members. To communicate with all those twenty old members respectively, *R* has to operate the key agreement protocol with them every one. In other words, in such a

design, the larger the number of members in a group one wishes to communicate with, the more resource and time will be needed to set up everything.

To solve this technical problem, we shall introduce a communication structure called the P2P-based online social networks. Following the concept, we will have the vehicles in the communication system play the roles of peers. In 2012, Yeh et al. proposed a framework for batch authentication and key agreement [40]. In their framework, a new user *R* can generate just one session key with the members of an old group. This reduces not only the computation cost but also the number of transmissions to be done. Details of Yeh et al.'s scheme will be in Section 3. Unfortunately, although Yeh et al.'s scheme comes in handy when only a few new users are to join an existing group, it does not seem capable of handling situations where groups join together, which is common practice in vehicular environments, for example a large vehicular team's members assembling in two different places and then joining together on the way. Therefore, to apply Yeh et al.'s design in VANET environments, we need to modify it a little bit so that cases of group integration can be properly taken care of.

In this section, we review the Yeh et al.'s scheme and compare it with traditional key agreement protocol and propose an improvement to let the approach become widely. The rest of the subsections are organized as follows. Subsection 4.2 reviews Yeh et al.'s scheme and compares it with traditional key agreement protocol in Subsection 4.3. After that, we present an improved scheme in Subsection 4.4 and the analysis of the multi key agreement scheme in Subsection 4.5. Finally, the conclusion is shown in Section 4.6.

4.2 Review of the Yeh et al.'s Scheme

First of all, the notations used in Yeh et al.'s scheme are shown in Table 4.2.1. In Yeh et al.'s scheme [40], there are three distinct protocols, namely the hash based protocol, the proxy-based protocol, and the certificate-based protocol, each for a different scenario. Because the three share the same basic principles, here we will only get to the details of the certificate-based protocol, which is designed especially to ensure the non-repudiation of a transaction. Interesting readers can refer [40].

When U_R wants to join U_A's group, the certificate-based protocol is started and the details are shown as follow.

(1) $U_R \rightarrow U_A : \{\{PK_A\{ID_R, N_R, UID = \{ID_1, ID_2, \dots, ID_{|\hat{U}|}\}, MAC_R\}$

i. U_R chooses a nonce N_R, and lists $UID = \{ID_1, ID_2, \dots, ID_{|\hat{U}|}\}$.

ii. U_R encrypts $\{ID_R, N_R, UID\}$ by U_A's public key PK_A.

iii. U_R computes $MAC_R = H(PK_A\{ID_R, N_R, UID\}, K_{RA} + N_R)$.

(2) $U_A \rightarrow U_R: \{PK_R\{N_R + 1, \hat{T}\}, MAC_A\}$

i. U_A obtains $\{ID_R, N_R, UID\}$ by decrypting $PK_A\{ID_R, N_R, UID\}$ and checks MAC_R.

ii. U_A confirms each U_R's trust level and encrypts $\{N_R + 1, \hat{T}\}$ by U_R's public key PK_R, where $\hat{T} = \{T_1, T_2, \dots, T_{|\hat{U}|}\}$.

iii. U_A computes $MAC_A = H(PK_R\{N_R + 1, \hat{T}\}, K_{RA} + N_R)$.

Table 4.2.1 Notations of the Yeh et al.'s scheme

Notation	Description
q, p	Large primes such that $p = 2q + 1$
g	The primitive root of prime q
RK_i	The private key of user U_i
PK_i	The public key of U_i. PK_i is used for ElGamal encryption such that $PK_i = g^{RK_i} \bmod p$
B_i	Representing n positive integers that are pairwise relatively primes used in CRT.
Requester (U_R)	A user who requests batch authentication.
Authenticator (U_A)	A user who assists U_R for the batch authentication.
$\hat{G}$	The set of all participants involved in the batch authentication. $\hat{G} = \{U_R, U_A, U_1, U_2, \ldots U_n\}$
$\lvert\hat{G}\rvert$	The number of all participants involved in the batch authentication
$\hat{U}$	A user group to be authenticated, $\hat{U} = \hat{G} - \{U_A, U_R\} = \{U_1, U_2, \ldots U_n\}$
$\lvert\hat{U}\rvert$	Representing n positive integers that are pairwise relatively primes used in CRT.
UID	The set of $\hat{U}$'s identities in this batch authentication session $UID = \{ID_1, ID_2, \ldots, ID_n\}$
N_i	A nonce picked by U_i
S	A random number serving as a seed of ElGamal proxy encryption key
T_i	The U_i's certificate
$\hat{T}$	The set of $\hat{U}$'s certificates $\hat{T} = \{T_1, T_2, \ldots T_n\}$
KP_R	The set of key parameters sent from U_R to $\hat{U}$ for key agreement. $KP_R = \{g^{m_1}, g^{m_2}, \ldots g^{m_n}\}$
$KP_{\hat{U}}$	The set of key parameters sent from $\hat{U}$ to U_R. $KP_{\hat{U}} = \{g^{n_1}, g^{n_2}, \ldots g^{n_n}\}\}$
$QR_{R,A}$	The authentication request message transmitted from U_R to U_A.
CR	The chain-reply messages passed through users in a user group.
MR	The reply messages for mutual authentication.

(3) $U_R \rightarrow U_1$:$\{C1, X, MAC_A\}$

i. U_R decrypts ciphertext and checks MAC_A.

ii. U_R computes $C1 = g^r \bmod p$, r is a random number, $r \in Z_q^*$.

iii. U_R computes

$$\xi_i = \{\text{trust level}, S, \delta = Sign(RK_R, C1), T_R, g^{m_i}\}$$

$$V_i = C2_i = \xi_i (PK_i)^r = \xi_i (g^{RK_i})^r$$

, where S is a random number used as the seed of ElGamal proxy encryption key and δ is a signature generated by Shamir-Tauman signature protocol [35].

iv. U_R accommodates V_i in message X by using CRT.

v. U_R computes $MAC_R = H(C1, X, S)$.

(4) $U_1 \rightarrow U_2 : \{C1, C2'_1, X, KP_{\text{ID}}, MAC_1\}$

i. U_1 obtains $V_1 (=C2_1)$ by calculating X mod B_1.

ii. U_1 computes

$$C2_1 \cdot C1^{-RK_1}$$

$$= \xi_1 (g^{RK_1})^r \cdot (g^r)^{-RK_1} \bmod p$$

$$= \xi_1$$

$$= \{\text{trust level}, S, \delta = Sign(RK_R, C1), T_R, g^{m_1}\}$$

by using his private key RK_1, and checks MAC_R.

iii. U_1 selects a random number n_1 and adds the key parameter g^{n_1} to KP_{ID} and computes the session key $SK_{R1} = (g^{m_1})^{n_1} \bmod p$.

iv. U_1 computes $C2'_1 = \xi' (PK_R)^{S+1}$, where $\xi' = \{ID_R\}$.

v. U_1 computes $MAC_1 = H(C1, C2'_1, X, KP_{\hat{U}}, S)$.

(5) $U_i \rightarrow U_{i+1}$ or $U_{|\hat{U}|} \rightarrow U_R : \{C1, C2'_{i-1}, X, KP_{\hat{U}} = \{g^{n_1}, g^{n_2}, \dots, g^{n_i}\}, MAC_i\}$

i. U_i obtains $V_i(=C2_i)$ by calculating X mod B_i.

ii. U_i decrypts ξ_i by computing $\xi_i = C2_i \cdot C1^{-RK_i}$, and checks MAC_{i-1} and δ.

iii. U_i selects a random number n_i and adds the key parameter g^{n_i} to $\mathrm{KP}_{\hat{U}}$ and computes the session key $SK_{Ri} = (g^{m_i})^{n_i} \bmod p$.

iv. U_i computes

$$C2'_i = C2'_{i-1} \cdot (PK_R)^{S+i} \bmod p$$

$$= \xi'(PK_R)^{\sum_{j=1}^{i}(S+j)} \bmod p$$

, where $\xi' = \{ID_R\}$.

v. U_i generates $MAC_i = H(C1, C2'_i, X, KP_{\hat{U}}, S)$.

(6) U_R

i. After receiving $:\{C1, C2'_{i-1}, X, KP_{\hat{U}} = \{g^{n_1}, g^{n_2}, \dots, g^{n_i}\}, MAC_i\}$ form $U_{|\hat{U}|}$, U_R checks $MAC_{|\hat{U}|}$ by S.

ii. U_R computes $C1' = (g)^{\sum_{j=1}^{|\hat{U}|}(S+j)} \bmod p$, and obtains ξ' by computing

$$C2'_{|\hat{U}|} \cdot (C1')^{-RK_R} \bmod p$$

$$= \xi'(PK_R)^{\sum_{j=1}^{|\hat{U}|}(S+j)} \times (g^{RK_R})^{-\sum_{j=1}^{|\hat{U}|}(S+j)} \bmod p$$

$$= \xi'(g^{RK_R})^{\sum_{j=1}^{|\hat{U}|}(S+j)} \times (g^{RK_R})^{-\sum_{j=1}^{|\hat{U}|}(S+j)} \bmod p$$

$$= \xi'$$

iii. If $\xi' = ID_R$, the user group $\widehat{U}$ is authenticated.

iv. U_R computes session keys $SK_{Ri} = (g^{n_i})^{m_i} \bmod p$ for each U_i, where $1 \leq i \leq |\hat{U}|$.

4.3 Analysis of Yeh et al.'s Scheme

In this subsection, we shall first compare Yeh et al.'s scheme with a traditional key agreement protocol and then point out a weakness of Yeh et al.'s scheme in operation.

4.3.1 Comparison with Traditional Key Agreement Protocol

In Yeh et al.'s scheme, batch key agreement is used to reduce the time and resources spent on message transference. According to Section VI of Yeh et al.'s paper [40] , in traditional ElGamal encryption, for *n* users to complete mutual authentication, as many as $2 \times \left(\frac{n \cdot (n-1)}{2}\right) = n^2 - n$ times of message transference are needed. The reason is that each user has to agree with all except for him-/herself, and traditional ElGamal encryption requires 2 times of message transference for a pair of users to do mutual authentication. Therefore, in case a new user U_R wants to join an old group U_A*'s group* of *n* members, then a total of *2n* times of message transference will be required if a traditional key agreement protocol is used, while the total number of message transferences can be cut down to *n+2* when Yeh et al.'s scheme is employed. Figure 4.3.1 and figure 4.3.2 illustrate how the two different protocols work, respectively. Due to the fact that messages

transmitted through wireless networks can easily be lost or intercepted, the number of message transferences had better be as small as possible.

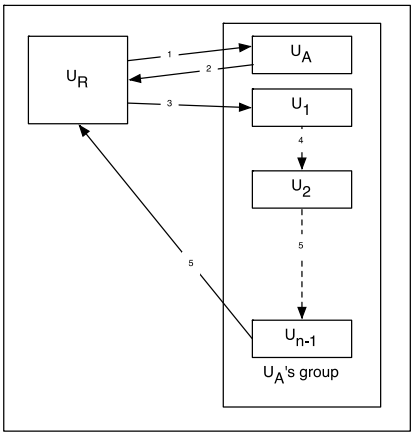


Figure 4.3.1 Message transference in traditional ElGamal encryption

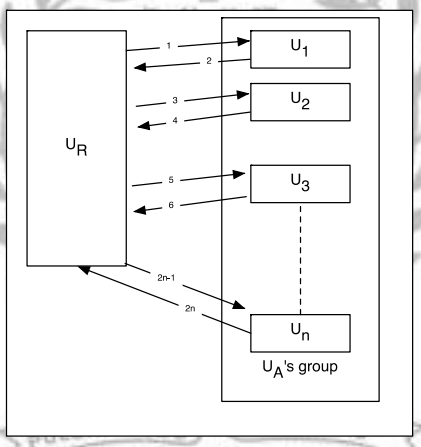


Figure 4.3.2 Message transference in Yeh et al.'s scheme

4.3.2 Leak of Yeh et al.'s Scheme

There is a leak in Yeh et al.'s scheme. When a new user wants to join an old group, the new user U_R transfers the message of key agreement to each member of the exited group U_A*'s group* by using an assembly value *X*, and each receiver U_i can obtain their unique message from *X* by calculating *X* mod B_i. To use CRT, U_R has to know each B_i and use them to generate the assembly value *X*. However, in Yeh et al.'s original paper, the source of the value B_i is not clearly stated; in other words, we have no idea whether

B_i is public or kept secret, and we do not know who owns it if it is kept secret. If B_i is public, then anyone can readily use it to obtain the secret message, which means the system security is completely broken. On the other hand, if B_i is kept secret, the U_R sure cannot obtain each B_i anyway. If U_R has to agree with each U_i about B_i before the key agreement protocol is even started, then we might as well have U_R and each U_i directly agree on their session key.

4.4 The Multi Key Agreement Scheme

To improve and further extend Yeh et al.'s scheme, we have made some modifications and applied the upgraded design in our proposed scheme for VANET environments. Assume there are two vehicular teams $U_R's\ group$ and $U_A's\ group$ that want to join together to form a bigger vehicular team. The members of $U_R's\ group$ are denoted as UR_0 and UR_j, where j=1, 2…m, and the members of $U_A's\ group$ are denoted as UA_0 and UA_i, where i=1, 2…n. In this case, all the vehicular members have previously agreed with each of their original teammates on the session key, and only UR_0 and UA_0 are from different groups but have agreed on the same session key. The notations are listed in Table 4.4.1.

4.4.1 Initiation

In this subsection, we will define the system public information. The public information issuer may be the manufacturer, system manager, or the certification authorities (CA).

(1) Let G be a cyclic additive group, and let G_T be a cyclic multiplicative group generated by P. G and G_T have the same prime order q, and |G|=|G_T|.

(2) Define $\hat{e}$: G×G→G_T be a bilinear map.

(3) Select two one-way hash function $h()$:{0,1}→Z_q^* and $H()$:{0,1}→Z_q^*.

(4) Select a random value s as the master key and generate the secret key $S_{ID} = s \cdot h(ID) \cdot P$.

(5) Publish {G, G_T, q, P, $h()$, $H()$} and pre-load the secret key on the OBU of each vehicle.

Table 4.4.1 Notations of the multi key agreement

Notation	Description
UA_0, UA_i	Members of $U_A's\ group$, i=1, 2…n
UR_0, UR_i	Members of $U_R's\ group$, j=1, 2…m
a_i	Random number chose by U_{Ai}
r_j	Random number chose by U_{Rj}
r, a	Random number
ID_X	ID of U_x
KAR	Session key between U_{A0} and U_{R0}
$KA_{x,y}$	Session key between U_{AX} and U_{AY}
$KR_{x,y}$	Session key between U_{RX} and U_{RY}
$E_x\{M\}$	Use x to symmetric encrypt the plaintext M
$D_x\{C\}$	Use x to symmetric decrypt the ciphertext C
T_1, T_2	Timestamp

4.4.2 Multi Establishing Session Key Protocol

When $U_R's\ group$ and $U_A's\ group$ want to combine to form a bigger vehicular team, UR_0 will contact UA_0 to reach an agreement on the session key *KAR* first. After that, the multi session key establishment protocol operates as follows.

(1) $UR_0 \rightarrow UR_1: \{KPR_0, CRP_0, T_1, MAC_{R0}, E_R(.)\}$

i. UR_0 selects two primes (p_R, q_R), generates the parameters $(N_R, \lambda_R, g_{R_1}, g_{R_2})$, and defines the encrypting function $E_R(.)$ and decrypting function $D_R(.)$ of the Paillier cryptosystems (see Section 2.3).

ii. UR_0 chooses a random number r_0 and computes r_0P.

iii. UR_0 generates a set $KPR_0 = \{r_0P\}$.

iv. UR_0 computes

$$CRP_0 = E_R\left(r_0 \cdot S_{ID_{R0}} \cdot T_1\right)$$

$$MAC_{R0} = H\left(ID_{R0}, KPR_0, CRP_0, KR_{0,1}, T_1, E_R(.)\right)$$

(2) $UR_j \rightarrow UR_{j+1}/UR_{m-1} \rightarrow UR_0: \{ID_{Rj}, CRP_j, MAC_{Rj}, T_1, E_R(.)\}$

i. UR_j checks T_1 and MAC_{Rj-1}.

ii. UR_j chooses a random number r_j and computes r_jP.

iii. UR_j generates a set $KPR_j = KPR_{j-1} \cup r_jP$.

iv. UR_j computes

$$CRP_j = CRP_{j-1} \cdot E_R\left(r_j \cdot S_{ID_{Rj}} \cdot T_1\right)$$

$$MAC_{Rj} = H(ID_{Rj}, KPR_j, CRP_j, KR_{j,j+1}, E_R(.))$$

(3) $UR_0 \rightarrow UA_0: \{ID_{R0}, EAR, MAC_{RA}\}$

i. UR_0 checks T_1 and MAC_{Rm-1}.

ii. UR_0 computes

$$CRP = D_R(CRP_m)$$

$$KPR = KPR_m$$

$$EAR = E_{KAR}\{ID_{R0}, IDR, \text{trust level}, CRP, KPR, T_1\}$$

$$MAC_{RA} = H(KAR, ID_{R0}, CRP, T_1, KPR)$$

(4) $UA_0 \rightarrow UA_1: \{ID_{A0}, X, EKPA, MAC_{A0}\}$

i. UA_0 decrypts EAR by KAR and checks T_1 and MAC_{RA}.

ii. UA_0 checks $\hat{e}(CRP, h(ID_{A0}) \cdot P)? =$

$\hat{e}\left(h(ID_{R0}) \cdot r_0 P, S_{ID_{A0}} \cdot T_1\right) \cdot \prod_{j=1}^{m} \hat{e}\left(h(ID_{Rj}) \cdot r_j P, S_{ID_{A0}} \cdot T_1\right)$.

iii. UA_0 generates the parameters and defines the encrypting function $E_A(.)$, decrypting function $D_A(.)$ of the Paillier cryptosystems as subsection 2.3.

iv. UA_0 chooses two random number $\{a, a_0\}$ and computes $a_0 P$.

v. UA_0 generates a set $KPA_0 = \{a_0 P\}$

$CAP_0 = E_A\left(a_0 \cdot S_{ID_{A0}} \cdot T_2\right)$, where T_2 is a timestamp,

$EKPA = E_{KA_{0,1}}\{KPA_0, CAP_0, T_2\}$.

vi. UA_0 computes VA_i and accommodates VA_i in message X by using CRT:

$$VA_i = \left\{ \begin{matrix} ID_{A0}, IDR, KPR, T_1, T_2, E_A(.), CRP \\ \hat{e}\left(\left(a + \sum_{l=1 \backslash l=i}^{n} KA_{0,l}\right) \cdot P, P\right) \end{matrix} \right\}$$

$X \equiv \sum_{i=1}^{n} \left(\frac{L}{h(KA_{0,i})} \times V_i \times A_i \right) \bmod L$

where $L = \prod_{i=1}^{n} h(KA_{0,i}), A_i \times \left(\frac{L}{h(KA_{0,i})} \right) \equiv 1 \bmod h(KA_{0,i})$.

vii. UA_0 computes

$MAC_{A,0} = H\left(X, KPA_0, CAP_0, T_1, T_2, \hat{e}\left(\left(a + \sum_{i=1}^{n} KA_{0,i}\right) \cdot P, P\right)\right)$.

(5) $UA_i \rightarrow UA_{i+1} / UA_n \rightarrow UA_0 : \{ID_{Ai}, X, EKPA, MAC_{Ai}\}$

i. UA_i decrypts $EKPA$ and obtains $VA_i = X \bmod A_i$.

ii. UA_i computes $\sigma = \hat{e}\left(\left(a + \sum_{l=1 \backslash l=i}^{n} KA_{0,l}\right)P, P\right) \cdot e(KA_{0,i} \cdot P, P)$.

iii. UA_i checks $\{T_1, T_2\}$ and

$MAC_{Ai-1} ? = H(X, KPA_{i-1}, T_1, T_2, CAP_{i-1}, \sigma)$

$\hat{e}(CRP, h(ID_{Ai}) \cdot P) ? =$

$\hat{e}\left(h(ID_{R0}) \cdot r_0 P, S_{ID_{Ai}} \cdot T_1\right) \cdot \prod_{j=1}^{m} \hat{e}\left(h(ID_{Rj}) \cdot r_j P, S_{ID_{Ai}} \cdot T_1\right)$

iv. UA_i chooses a random number a_i and computes $a_i P$.

v. UA_i computes

$KPA_i = KPA_{i-1} \cup a_i P$

$CAP_i = CAP_{i-1} \cdot E_A(a_i \cdot S_{ID_{Ai}} \cdot T_2)$

$EKPA = E_{KA_{i,i+1}}\{KPA_i, CAP_i\}$

$MAC_{Ai} = H(X, KPA_i, \sigma)$

(6) $UA_0 \rightarrow UR_0 : \{ID_{A0}, ERA, MAC_{AR}\}$

i. UA_0 decrypts $EKPA$ and

$MAC_{An} = ? H\left(X, KPA_n, CAP_n, T_1, T_2, \hat{e}\left(\left(a + \sum_{i=1}^{n} KA_{0,l}\right)P, P\right)\right)$.

ii. UA_0 computes $CAP = D_A(CAP_n)$ and $KPA = KPA_n$.

iii. UA_0 computes $ERA = E_{KAR}\{ID_{A0}, IDA = \{ID_{Ai}\}, KPA, CAP, T_1, T_2\}$

iv. UA_0 computes $MAC_{AR} = H(ERA, KPA, CAP, T_1, T_2)$

(7) $UR_0 \rightarrow UR_j$: (C_{Rj}, MAC_{R0})

i. UR_0 decrypts ERA by KAR and checks $\{T_1, T_2\}$.

ii. UR_0 checks computes MAC_{AR} and $\hat{e}(CAP, h(ID_{R0}) \cdot P)? =$

$$\hat{e}\left(h(ID_{A0}) \cdot a_0 P, S_{ID_{R0}} \cdot T_2\right) \cdot \prod_{i=1}^{n} \hat{e}\left(h(ID_{Ai}) \cdot a_i P, S_{ID_{R0}} \cdot T_2\right)$$

iii. UR_0 computes

$$C_{Rj} = E_{KR_{0,j}}\{ID_{R0}, ID_{A0}, IDA, KPA, CAP, T_1, T_2\}$$

$$MAC_{R0} = H(KPA, CAP, T_1, T_2)$$

(8) UR_j

i. UR_j decrypts C_{Rj} and checks MAC_{R0} and $\{T_1, T_2\}$.

ii. UR_j checks $\hat{e}\left((CAP)P, h\left(ID_{Rj}\right) \cdot P\right)? =$

$\hat{e}\left(h(ID_{A0}) \cdot a_0 P, S_{ID_{Rj}} \cdot T_2\right) \cdot \prod_{i=1}^{n} \hat{e}\left(h(ID_{Ai}) \cdot a_i P, S_{ID_{Rj}} \cdot T_2\right)$.

iii. UR_j computes the session key $K_{A_i B_j} = a_i b_j P$ with UA_i.

4.5 Analysis of the Multi Key Agreement Scheme

In this section, we shall show the results of our analysis of the proposed scheme. First, the BAN logic proposed by Burrows et al. [5] was used to confirm the correctness of the

proposed scheme. Then, we conducted a secrecy check. Finally, the proposed scheme was compared with Yeh et al.'s scheme in terms of performance.

4.5.1 Correctness Analysis of the Multi Key Agreement with BAN Logic

The BAN logic is a well-accepted method for correctness check of information exchange protocols [5, 6]. As a logic of belief and action, the BAN logic comprises a set of simple rules to help users determine whether the information exchanged is trustworthy or not. Before we can put the BAN logic is use, we must define the basic notations, goals, and assumptions first. Now let's analyze the protocol of the proposed scheme with the BAN logic.

(1) Notations

First of all, here are the syntax and notations of the BAN logic. Let's define A, B, X and Y as participator A, participator B, value X, and value Y, and then use some instances to show how the logic works [5].

i. $A|\equiv X$: A believes X is trust.

ii. $A|\equiv B$: A believes B's actions. For example: $A|\equiv B|\equiv X$ means that A believes B believes X is trust.

iii. $A \lhd X$: A sees or holds X.

iv. $A|\sim X$: A has once said the X in the process this time.

v. $\#(X)$: X is fresh, that means X is recent or X is a nonce.

vi. $A \overset{X}{\leftrightarrow} B$: X is a secret key shared between A and B.

vii. $\overset{x}{\mapsto} A$ and X^{-1}: $\overset{x}{\mapsto} A$ is the public key of A and X^{-1} is the privacy key of A.

viii. $\langle Y \rangle_X$: Plain text Y is combined with X, where X can be secret value in this rule.

ix. (X, Y): X or Y is one part of formula (X, Y).

x. $A|\Rightarrow X$: A has complete control over X. It can be used for denoting a certificate authority.

xi. $\frac{Rule\ 1}{Rule\ 2}$: We can infer *Rule* 1 from *Rule* 2. For example: $\frac{A \text{ creates } X}{A\ |\equiv \#(X)}$ means that because A creates X, A believes X is fresh.

For the sake convenience, we set i=1, 2…n and j=1, 2…m in following example.

(2) Goals

To check the correctness of the proposed scheme, we set four goals. If all four goals are achieved, that means we have good reasons to believe the protocol of the proposed scheme is correct. The participators in our protocol are the CA and the members of two different groups. The members of $UR's\ group$ are UR_0 and UR_j, and the members of $UA's\ group$ are UA_0 and UA_i. The major goal of the proposed scheme is to let the members from different groups exchange secret information so that they can establish a multiple session key. For this reason, we hope the proposed scheme can make those participators believe that correct targets said the exchanged information. The goals of the proposed scheme are stated in the language of the BAN logic as follows.

i. UA_0 and $UA_i|\equiv UR_0|\sim r_0 P$

ii. UA_0 and UA_i $|\equiv UR_j|\sim r_j P$

iii. UR_0 and $UR_j|\equiv UA_0|\sim a_0 P$

iv. UR_0 and $UR_j|\equiv UA_i|\sim a_i P$

(3) Assumptions

To analyze the multi key agreement protocol, the following assumptions need to be established:

i. $UR_0|\equiv UR_0 \overset{KR_{j,0}}{\longleftrightarrow} UR_j$

ii. UR_j $|\equiv UR_0 \overset{KR_{j,0}}{\longleftrightarrow} UR_j$

iii. UR_j $|\equiv UR_{j-1} \overset{KR_{j-1,j}}{\longleftrightarrow} UR_j$

iv. $UR_0|\equiv UR_0 \overset{KAR}{\longleftrightarrow} UA_0$

v. $UA_0|\equiv UR_0 \overset{KAR}{\longleftrightarrow} UA_0$

vi. $UA_0|\equiv UA_0 \overset{KA_{0,i}}{\longleftrightarrow} UA_i$

vii. $UA_i|\equiv UA_0 \overset{KA_{0,i}}{\longleftrightarrow} UA_i$

viii. $UA_i|\equiv UA_{i-1} \overset{KA_{i-1,i}}{\longleftrightarrow} UA_i$

ix. $UR_0, UR_j, UA_0, UA_i|\equiv CA|\Longrightarrow s$

(4) Correctness analysis of the multi key agreement scheme's verification

In this subsection, we analyze the correctness of the multi key agreement scheme with the BAN logic. The details are as follows:

Message 1: $UR_0 \rightarrow UR_1/UR_j \rightarrow UR_{j+1}/UR_m \rightarrow UR_0$:

$$\langle\{r_{0,1,\dots j}\cdot P\},\{r_{0,1,\dots j}\cdot s\cdot T_1\cdot h(ID_{Rj})P\},T_1\rangle_{KR_{j,j+1}}$$

i. $$\frac{UR_j|\equiv UR_{j-1}\overset{KR_{j-1,j}}{\longleftrightarrow}UR_j}{UR_j|\equiv UR_{j-1}|\sim(r_{j-1}\cdot s\cdot h(ID_{R0})P,T_1)}$$

ii. $$\frac{UR_j|\equiv UR_{j-1}|\sim(r_{j-1}\cdot s\cdot h(ID_{R(j-1)})P,T_1),\ UR_j\triangleleft T_1}{UR_j|\equiv\#(r_{j-1}\cdot s\cdot h(ID_{R(j-1)})P,T_1)}$$

Message 2: $UR_0 \rightarrow UA_0$:

$$\langle\{r_{0,1,\dots m}\cdot P\},\{r_{0,1,\dots m}\cdot s\cdot T_1\cdot h(ID_{R0})P\},T_1\rangle_{KAP}$$

iii. $$\frac{UA_0|\equiv UR_0\overset{KAR}{\longleftrightarrow}UA_0}{UA_0|\equiv UR_0|\sim(\{r_{0,1,\dots m}\cdot P\},\{r_{0,1,\dots m}\cdot s\cdot T_1\cdot h(ID_{R0,1,\dots m})P\},T_1)}$$

iv. $$\frac{UA_0|\equiv UR_0|\sim(\{r_{0,1,\dots m}\cdot P\},\{r_{0,1,\dots m}\cdot s\cdot T_1\cdot h(ID_{R0,1,\dots m})P\},T_1),UR_j\triangleleft T_1}{UA_0|\equiv\#(\{r_{0,1,\dots m}\cdot P\},\{r_{0,1,\dots m}\cdot s\cdot T_1\cdot h(ID_{R0,1,\dots m})P\},T_1)}$$

v. $$\frac{UA_0|\equiv\#(\{r_{0,1,\dots m}\cdot P\},\{r_{0,1,\dots m}\cdot s\cdot T_1\cdot h(ID_{R0,1,\dots m})P\},T_1),UA_0\triangleleft s\cdot h(ID_{A0})P,UA_0|\equiv CA|\Longrightarrow s}{UA_0\ |\equiv UR_0|\sim r_0P,UA_0\ |\equiv UR_j|\sim r_jP}$$

Message 3: $UA_0 \rightarrow UA_1/UA_1 \rightarrow UA_{i+1}/\ UA_n \rightarrow UA_0$:

$$\begin{pmatrix}\langle\{r_{0,1,\dots m}\cdot P\},\{r_{0,1,\dots m}\cdot s\cdot T_1\cdot h(ID_{R0,1,\dots m})P\},T_1,T_2\rangle_{KA_{0,i}},\\ \langle\{a_{0,1,\dots i}\cdot P\},\{a_{0,1,\dots i}\cdot s\cdot T_2\cdot h(ID_{A0})P\},T_2\rangle_{KA_{i,i+1}}\end{pmatrix}$$

vi. $$\frac{UA_i|\equiv UR_0\overset{KA_{0,i}}{\longleftrightarrow}UA_i}{UA_i|\equiv UA_0|\sim(\{r_{0,1,\dots m}\cdot P\},\{r_{0,1,\dots m}\cdot s\cdot T_1\cdot h(ID_{R0,1,\dots m})P\},T_1,T_2)}$$

vii. $$\frac{UA_i|\equiv UA_0|\sim(\{r_{0,1,\dots m}\cdot P\},\{r_{0,1,\dots m}\cdot s\cdot T_1\cdot h(ID_{R0,1,\dots m})P\},T_1,T_2),UA_i\triangleleft T_1}{UA_i|\equiv\#(\{r_{0,1,\dots m}\cdot P\},\{r_{0,1,\dots m}\cdot s\cdot T_1\cdot h(ID_{R0,1,\dots m})P\},T_1)}$$

viii. $$\frac{UA_0|\equiv\#(\{r_{0,1,\dots m}\cdot P\},\{r_{0,1,\dots m}\cdot s\cdot T_1\cdot h(ID_{R0,1,\dots m})P\},T_1),UA_i\triangleleft s\cdot h(ID_{Ai})P,UA_i|\equiv CA|\Rightarrow s}{UA_i\ |\equiv UR_0|\sim r_0P,UA_i\ |\equiv UR_j|\sim r_jP}$$

Message 4: $UA_0 \rightarrow UR_0$:

$$\langle\{a_{0,1,\dots n}\cdot P\},\{a_{0,1,\dots n}\cdot s\cdot T_2\cdot h(ID_{A0,1,\dots n})P\},T_1,T_2\rangle_{KAR}$$

ix. $$\frac{UA_0|\equiv UR_0\overset{KAR}{\longleftrightarrow}UA_0}{UR_0|\equiv UA_0|\sim(\{a_{0,1,\dots n}\cdot P\},\{a_{0,1,\dots n}\cdot s\cdot T_2\cdot h(ID_{A0,1,\dots n})P\},T_1,T_2)}$$

x. $$\frac{UR_0|\equiv UA_0|\sim(\{a_{0,1,\dots n}\cdot P\},\{a_{0,1,\dots n}\cdot s\cdot T_2\cdot h(ID_{A0,1,\dots n})P\},T_1,T_2),UR_0\triangleleft(T_1,T_2)}{UR_0|\equiv\#(\{a_{0,1,\dots n}\cdot P\},\{a_{0,1,\dots n}\cdot s\cdot T_2\cdot h(ID_{A0,1,\dots n})P\},T_1,T_2)}$$

xi. $$\frac{UR_0|\equiv\#(\{a_{0,1,\dots n}\cdot P\},\{a_{0,1,\dots n}\cdot s\cdot T_2\cdot h(ID_{A0,1,\dots n})P\},T_1,T_2),UR_0\triangleleft s\cdot h(ID_{R0})P,UR_0|\equiv CA|\Rightarrow s}{UR_0\ |\equiv UA_0|\sim a_0P,UR_0\ |\equiv UA_i|\sim a_iP}$$

Message 5: $UR_0 \rightarrow UR_j$:

$$\langle\{a_{0,1,\dots n}\cdot P\},\{a_{0,1,\dots n}\cdot s\cdot T_2\cdot h(ID_{A0,1,\dots n})P\},T_1,T_2\rangle_{KR_{0,j}}$$

xii. $$\frac{UR_j|\equiv UR_0\overset{KR_{0,j}}{\longleftrightarrow}UR_j}{UR_j|\equiv UR_0|\sim(\{a_{0,1,\dots n}\cdot P\},\{a_{0,1,\dots n}\cdot s\cdot T_2\cdot h(ID_{A0,1,\dots n})P\},T_1,T_2)}$$

xiii. $$\frac{UR_j|\equiv UR_0|\sim(\{a_{0,1,\dots n}\cdot P\},\{a_{0,1,\dots n}\cdot s\cdot T_2\cdot h(ID_{A0,1,\dots n})P\},T_1,T_2),UR_j\triangleleft(T_1,T_2)}{\mathrm{U}R_j|\equiv\#(\{a_{0,1,\dots n}\cdot P\},\{a_{0,1,\dots n}\cdot s\cdot T_2\cdot h(ID_{A0,1,\dots n})P\},T_1,T_2)}$$

xiv. $$\frac{UR_j|\equiv\#(\{a_{0,1,\dots n}\cdot P\},\{a_{0,1,\dots n}\cdot s\cdot T_2\cdot h(ID_{A0,1,\dots n})P\},T_1,T_2),UR_j\triangleleft s\cdot h(ID_{Rj})P,UR_j|\equiv CA|\Rightarrow s}{UR_j\ |\equiv UA_0|\sim a_0P,UR_j\ |\equiv UA_i|\sim a_iP}$$

Finally, we can infer that the multi key agreement scheme really achieves the original goals from formula v, viii, xi, and xiv. For this reason, we claim the multi key agreement scheme's protocol is correct.

4.5.2 Security Analysis of the Multi Key Agreement Scheme

In this subsection, we will discuss the proposed scheme in terms of some secrecy issues that are frequently mentioned as far as key agreement protocols are concerned. The

most critical security issues about key agreement for multi users include [13, 16, 19, 21, 24, 26] forward secrecy/backward secrecy, unknown key-share resilience, known session-specific temporary information secrecy, and collusion attack resistance. Therefore, we will cover those issues as follows.

(1) Forward secrecy/Backward secrecy

Perfect forward secrecy is said to be achieved when the long-term private keys of one or more of the entities are disclosed but the secrecy of previously established session keys still hold. Similarly, perfect backward secrecy means that a disclosed secret key reveals no information about the session keys that follows it. In the proposed scheme, each session key $K_{A_iB_j}$ is composed of two random numbers, namely $r_{0,1\ldots,m}$ and $a_{0,1\ldots,n}$. The long-term private key $s \cdot h(ID) \cdot P$ of each user is only used to aid the user confirm the validity of his/her target. For this reason, the sessions are always secret even if the master key s is compromised. Hence, we can claim with confidence that the proposed scheme does achieve perfect forward secrecy and perfect backward secrecy.

(2) Known session-specific temporary information secrecy

During the process of session key generation, the participator will select some private information to randomize the session key, and the random private information should be kept secret so that the session key generated will not be compromised. In the proposed scheme, we use r_0, r_j, a_0, and a_i to do the job of randomization. According to ECDLP (refer to Section 2.1), it is extremely difficult to find r_0, r_j, a_0, and a_i even if $r_0 \cdot P, r_j \cdot P, a_0 \cdot P$, and $a_i \cdot P$ are known.

(3) Unknown key-share resilience

This is the most crucial issue for key agreement protocols. In an attack, the adversary can impersonate the key agreement target between two entities when they are exchanging secret information for key agreement, and this attack is well known as the man-in-the-middle attack (MITM). In the proposed scheme, a large number of entities are involved in the key agreement process with many secret messages transferred from one entity to another. Therefore, there is a high risk of being attacked by MITM. For this reason, how to resist MITM is a key point.

The success of MITM is built on the basis that the adversary can replace the original secret message with some fake data without being detected because the legal entities cannot correctly verify the source of the information. In the case of the proposed scheme, the adversary may be an outsider or an insider. We will discuss both possibilities.

i. The adversary is an outsider

In fact, it is difficult for an outsider to launch a MITM attack on the proposed scheme because all secret information transferred is protected by secret channels, i.e. the session keys $KAR, KR_{j,j+1}$, $KA_{i,i+1}$, etc. For this reason, we claim that an outsider cannot crack the proposed scheme by using MITM.

ii. The adversary is a participant

In the proposed scheme, during the operation of the key agreement protocol, an inside entity can obtain information from another entity. If the inside confidentiality of the protocol was not strong enough, the system would be vulnerable to attacks from inside. To make sure that inside attacks can do no harm, we use two mechanisms to ensure the secrecy of the information transferred between entities. The first is homomorphism

encryption. Each general entity, i.e. UR_j and UA_i, can use the homomorphism encryption function to encrypt his/her authentication information $r_j \cdot S_{ID_{Rj}} \cdot T_1$ and $a_i \cdot S_{ID_{Ai}} \cdot T_2$ to ensure that his/her exchanged information $r_j P$ and $a_i P$ cannot be replaced.

Due to the protective shield formed by the homomorphism encryption function, only the agreement entities, i.e. UR_0 and UA_0, can decrypt the ciphertext {*CRP*, *CAP*} and obtain the aggregate authentication information. However, what if the agreement entities are malicious? In the proposed scheme, the aggregate authentication information, which forms the second protective mechanism against inside attacks, is the sum of the authentication information. It is extremely difficult to derive the unknown unique value $r_j \cdot S_{ID_{Rj}} \cdot T_1$ from $r_0 \cdot S_{ID_{R0}} \cdot T_1 + \sum_{j=1}^{m} \left(r_j \cdot S_{ID_{Rj}} \cdot T_1 \right)$ or to figure out $a_i \cdot S_{ID_{Ai}} \cdot T_2$ from $a_0 \cdot S_{ID_{A0}} \cdot T_2 + \sum_{i=1}^{n} \left(r_i \cdot S_{ID_{Ai}} \cdot T_2 \right)$. Hence, we know that the agreement entities have no way to replace the authentication information $r_j \cdot S_{ID_{Rj}} \cdot T_1$ and $a_i \cdot S_{ID_{Ai}} \cdot T_2$ as general entities. Although an adversary from inside can modify the authentication information $r_j \cdot S_{ID_{Rj}} \cdot T_1$ and $a_i \cdot S_{ID_{Ai}} \cdot T_2$, another user can readily notice the modification by checking the authentication information. Therefore, we claim that the MITM attack will take no effect on the proposed scheme.

(4) Collusion attack resistance

In the proposed scheme, the exchanged information is transferred via multi users. If the proposed scheme was vulnerable to the collusion attack, the malicious users would be able to ally to decrypt the encrypted information and modify it. To rule out that possibility, we use the secret key S_{ID} and the homomorphism encryption to protect the exchanged information. Without homomorphism encryption, a malicious user could derive and replace the exchanged data by computing the variations of the authentication information

as follows. Suppose users UR_x and UR_{x+2} are malicious allies and the transferred authentication information is not encrypted. UR_x can compute $M_1 = r_0 \cdot S_{ID_{R0}} \cdot T_1 + \sum_{j=1}^{x} \left(r_j \cdot S_{ID_{Rj}} \cdot T_1 \right)$, and UR_{x+2} can obtain the information $M_2 = r_0 \cdot S_{ID_{R0}} \cdot T_1 + \sum_{j=1}^{x+1} \left(r_j \cdot S_{ID_{Rj}} \cdot T_1 \right)$ from UR_{x+1}. After that, the malicious allies can use $M_2 - M_1$ to derive UR_{x+1}'s authentication information $r_{x+1} \cdot S_{ID_{Rx+1}} \cdot T_1$ and replace it. With homomorphism encryption in the way, the malicious allies will have no way to know the plaintext of the authentication information, and nor can the specific value $r_j \cdot S_{ID_{Rj}} \cdot T_1$ or $a_i \cdot S_{ID_{Ai}} \cdot T_1$ be figured out because the master key s is kept secret. Therefore, we claim that the proposed scheme can resist the collusion attack.

4.5.3 Performance Analysis of the Multi Key Agreement Scheme

In this subsection, we will discuss the performance of the proposed scheme. Earlier in section 4.3.1, we mentioned that Yeh et al.'s scheme could significantly reduce the number of information transmissions [40]. However, Yeh et al.'s scheme can handle cases where only one new user is to join an old group (i.e., 1 vs. n) but will be overwhelmed when two large groups are to join together (i.e., n vs. m). In addition, as we pointed out in section 4.3.2, Yeh et al.'s scheme also has a flaw. By contrast, in the proposed scheme, we have not only fixed the defect but also extended the field of application from one new comer joining an existing group to group integration. Let’s take an example to help visualize the improvement offered by the proposed scheme. In the same instance we brought up earlier in section 4.4, there are two vehicular teams $U_R's\ group$ and $U_A's\ group$ intending to combine to form a bigger team. The members of $U_R's\ group$

include UR_0 and UR_j, where *j*=1, 2…m, and $U_A's\ group$ comprises UA_0 and UA_i, where *i*=1, 2…n. In a traditional one-on-one scheme, the members should operate the key agreement protocol $(m+1)\cdot(n+1) = m\cdot n + m + n + 1$ times. If the two teams are roughly on the same scale, then the operation times grow exponentially. Besides, at least 2 message transferences will have to be done when a traditional protocol is operated once. That means the traditional protocol needs $2(m\cdot n + m + n + 1)$ times of message transference to complete this work. In Yeh et al.'s scheme, the transference times can reduce to $m(n+2)$, taking the case for m new users joining the existing team $U_A's\ group$ of n members. Although Yeh et al.'s scheme can reduce the transference times by almost a half, there is still an exponential growth. In the proposed scheme, the message transference times are further significantly reduced to only $2m + n + 2$, and the growth is linear. As one of the major characteristics of VANET is that the nodes are not fixed or limited to a small area, bigger numbers of message transferences mean higher potential risks of loss of contact or secret information leakage, and of course the proposed scheme is the best of its kind in this respect because it requires the least message transferences. In addition, the first couple of steps of the proposed protocol can easily be adapted to further broaden the field of application. Therefore, the proposed scheme is obviously the best design currently available for the establishment of multiple session keys. Figure 4.5.1 and figure 4.5.2 show the flows of different schemes in this explain, and table 4.5.1 demonstrate the comparison of transmission times between different schemes.

For the characteristic of VANET, the node is not fixed or limited to a small area. Hence, more message transference times mean potential risks, such as contact broken. In addition, the step 1 and step 2 of the multi key agreement scheme even can be well

advance. Therefore, we claim the multi key agreement scheme is the best suitable for multiple establishing session keys.

Table 4.5.1 Comparison of transmission times

	1 to 1	1 to $U_A's\ group$	$U_R's\ group$ to $U_A's\ group$
Traditional one by one scheme	2	$2(n+1)$	$2(n+1)(m+1)$
The Yeh et al.'s scheme		$(n+1)+2$	$(m+1)(n+1+2)$
The multi key agreement scheme			$2m+n+2$

*There are $(n+1)$ members in $U_A's\ group$.
*There are $(m+1)$ members in $U_R's\ group$.

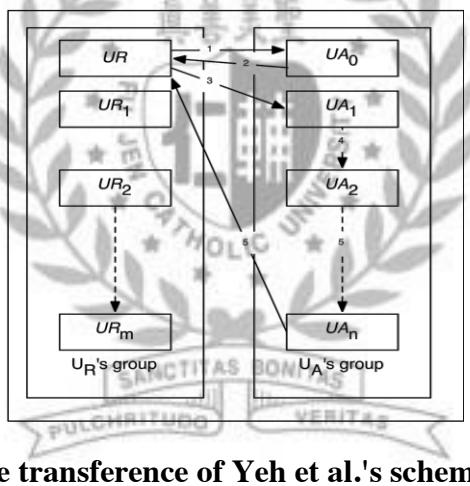


Figure 4.5.1 Message transference of Yeh et al.'s scheme in groups combined

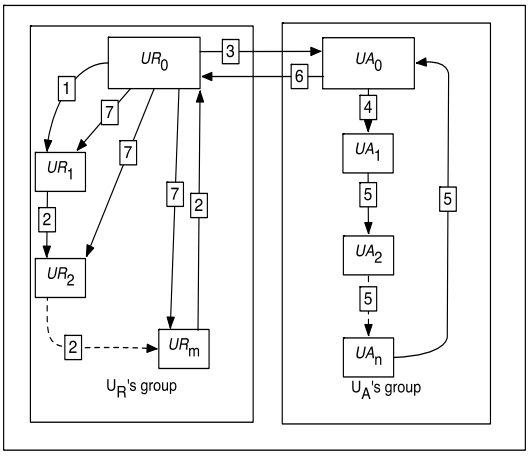


Figure 4.5.2 Message transference of the multi key agreement scheme in groups combined

Chapter 5 Conclusions

With the communication technology development, VANET can be regarded as a "predictable" technology. In this study, we focus on two different applications of VANET, batch verification for V2R and multi key agreement for V2V. In V2R, we proposed an improved batch verification scheme for VANET public commutation in chapter 3. The batch verification scheme overcomes the weaknesses of Zhang et al.'s scheme and improves the efficiency specifically, and hoped that the scheme can enhance the quality of traffic.

In V2V, we proposed a novel formwork to operate the multi-key agreement. Because VANET deals with mobile networks where moving vehicles are used as nodes, the grouping status and position of the users are subject to change. Under such circumstance, our focus when trying to create a suitable key agreement protocol is to reduce the necessary times of secret information exchange so as to minimize security risk. In this thesis, we have not only pointed out and mended a defect of the Yeh et al.'s scheme but have also extended its field of application from cases of *1* vs. *n* to cases of *n* vs. *m*. The correctness of the proposed scheme has been verified by the BAN-logic, and the secrecy of the proposed scheme has also been confirmed as various possible attacks have been proven ineffective. Judged by performance, the proposed scheme is by far the best system for the use in VANET environments. In fact, the proposed scheme is not merely very suitable for VANET setups but can also be used to combine different social community networks or platforms. In the future, we will further develop the features of batch scheme for VANET, such as the identifying illegal signatures, to design new schemes in order to gain more efficiency.

References

[1] T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless Secret Key Generation Exploiting Reactance-Domain Scalar Response of Multipath Fading Channels," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 11, pp. 3776-3784, 2005.

[2] ASTM E2213 - 03(2010) Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems 8212; 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ASTM.org, form : http://www.astm.org/Standards/E2213.htm, accessed: 2013/07/05.

[3] Dan Boneh and Matt Franklin, "Identity-Based Encryption from the Weil Pairing," *Lecture Notes in Computer Science*, vol. 2139, pp. 213-229, 2001.

[4] Azzedine Boukerche, Horacio A.B.F. Oliveira, Eduardo F. Nakamura, and Antonio A.F. Loureiro "Vehicular Ad Hoc Networks: A New Challenge for Localization-Based Systems," *Computer Communications*, vol. 31, no. 12, pp. 2838-2849, 2008.

[5] M. Burrows, M. Abadi, and R. Needham, "A Logic of Authentication," *ACM Transactions Computer Systems*, vol. 8, no. 1, pp. 18-36, 1990.

[6] Chin-Chen Chang and Chia-Yin Lee, "A secure single sign-on mechanism for distributed computer networks," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 1, pp. 629-637, 2012.

[7] Liqun Chen, Siaw-Lynn Ng, and Guilin Wang , "Threshold Anonymous Announcement in VANETs," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 605-615, 2011.

[8] T. W. Chim, S. M. Yiu, Lucas C. K. Hui, and Victor O. K. Li, "SPECS: Secure and Privacy Enhancing Communications Schemes for VANETs," *Ad Hoc Network*s, vol. 9, no. 12, pp. 189-203, 2011.

[9] Arzoo Dahiya and Dr. R. K. Chauhan, "A Comparative Study of MANET and VANET Environment," *Journal of Computing*, vol. 2, no. 7, pp. 87 - 92, 2010.

[10] Whitfield Diffie and Martin E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, nol. 6, pp. 644-654, 1976.

[11] Amos Fiat, "Batch RSA," *Lecture Notes in Computer Science*, vol. 435, no. 17, pp. 175-185, 1990.

[12] Mainak Ghosh, Anitha Varghese, Arobinda Gupta, Arzad A. Kherani, and Skanda N. Muthaiah, "Detecting Misbehaviors in VANET with Integrated Root-cause Analysis," *Ad Hoc Networks*, vol. 8, no. 7, pp. 778-790, 2010.

[13] Mengbo Hou and Qiuliang Xu, "An Efficient and Secure One-Round Authenticated Key Agreement Protocol without Pairings," in *Proceeding 2011 International Conference on Multimedia Technology (ICMT)*, pp. 160-163, 2011.

[14] Jiun-Long Huang, Lo-Yao Yeh, and Hung-Yu Chien, "ABAKA: An Anonymous Batch Authenticated and Key Agreement Scheme for Value-Added Services in Vehicular Ad Hoc Networks," *IEEE Transaction on Vehicular Technology*, vol. 60, no. 1, pp. 248-262, 2011.

[15] Jean-Pierre Hubaux, Srdjan CApkun, and Jun Luo, "The Security and Privacy of Smart Vehicles," *IEEE Security & Privacy*, vol. 2, no. 3, pp. 49 - 55, 2004.

[16] Min-Shiang Hwang, Chih-Wei Lin, Cheng-Chi Lee, "Improved Yen-Joye's Authenticated Multiple-key Agreement Protocol," *IEE Electronics Letters*, vol. 38, no. 23, pp. 1429-1431, 2002.

[17] Subhash Kak, "Computational Aspects of the Ãryabhata Algorithm," *Indian Journal of History of Science*, vol. 21, no. 1, pp. 62-71, 1986.

[18] Cheng-Chi Lee, "A Simple Key Agreement Scheme Based on Chaotic Maps for VSAT Satellite Communications," *International Journal of Satellite Communications and Networking*, vol. 31, no.4, pp. 177-186, 2013.

[19] Cheng-Chi Lee, Chin-Ling Chen, Hsia-Hung Ou, Lung Albert Chen, "Extension of an Efficient 3GPP Authentication and Key Agreement Protocol," *Wireless Personal Communication*s, vol. 68, no. 3, pp. 861-872, Feb. 2013.

[20] Cheng-Chi Lee, Chun-Ta Li, Te-Yu Chen, Chia-Ying Wu, "Towards Secure User Authentication and Key Agreement Protocol Based on Bilinear Pairings for Mobile Client-Server Environments," accepted and to appear in *Telecommunication Systems*.

[21] Cheng-Chi Lee, Chun-Ta Li, Kou-You Huang, Shiow-Yuan Huang, "An Improvement of Remote Authentication and Key Agreement Schemes," *Journal of Circuits, Systems, and Computers*, vol. 20, no. 4, pp. 697-707, 2011.

[22] Wenmin Li, Qiaoyan Wen, Qi Su, and Zhengping Jin, "An Efficient and Secure Mobile Payment Protocol for Restricted Connectivity Scenarios in Vehicular Ad Hoc Network," *Computer Communications*, vol. 35, no. 2, pp. 188-195, 2011.

[23] Xiaodong Lin, Xiaoting Sun, Pin-Han Ho, and Xuemin Shen, "GSIS: A Secure and Privacy-Preserving Protocol for Vehicular Communications," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3442 - 3456, 2007.

[24] Chen Li-Qing and Hu Rong-lin, "Group Key Agreement Scheme for Mobile Ad Hoc Networks Based on Threshold Secret Sharing," in *Proceeding 2010 Third International Symposium on Electronic Commerce and Security (ISECS)*, pp. 176-180, 2010.

[25] Atsuko Miyaji, Masaki Nakabayashi, and Shunzou Takano, "New Explicit Conditions of Elliptic Curve Traces for FR-Reduction," *IEICE Transaction on Fundamentals of Electronics*, vol. E84-A, no. 5, pp. 1234-1243, 2001.

[26] R. Mokhtarnameh, , Sin Ban Ho, and N. Muthuvelu , "An Enhanced Certificateless Authenticated Key Agreement Protocol," in *Proceeding 2011 13th International Conference on Advanced Communication Technology (ICACT)*, pp. 802-806, 2011.

[27] K. Muthumayil, V. Rajamani, S. Manikandan, and M. Buvana, "A Group Key Agreement Protocol Based on Stability and Power Using Elliptic Curve Cryptography," in *Proceeding 2011 International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT)*, pp. 1051-1056, 2011.

[28] P. Paillier, "Public-key Cryptosystems Based on Composite Degree Residuosity Classes," in *Proceeding EUROCRYPT*, pp. 223-238, 1999.

[29] Esther Palomar, José M. de Fuentes, Ana I. González-Tablas, and Almudena Alcaide, "Hindering False Event Dissemination In VANETs With Proof-Of-Work Mechanisms," *Transportation Research Part C: Emerging Technologies*, vol. 23, pp. 85-97, 2012.

[30] Maxim Raya and Jean-Pierre Hubaux, "Securing Vehicular Ad hoc Networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39-68, 2007.

[31] David Antolino Rivas, José M. Barceló-Ordinas, Manel Guerrero Zapata, Julián D. Morillo-Pozo, "Security On Vanets: Privacy, Misbehaving Nodes, False Information and Secure Data Aggregation," *Journal of Network and Computer Applications*, vol. 34, no. 6, pp. 1942-1955, 2011.

[32] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol.21, no.2, pp. 120-126, 1978.

[33] S. Ruj and A. Nayak, "A Decentralized Security Framework for Data Aggregation and Access Control in Smart Grids," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 196-205, 2013.

[34] Mike Scott, "Efficient Implementation of Cryptographic Pairings," [Online]. Available: ftp://ftp.disi.unige.it/pub/person/MoraF/CRYPTO/PARING/mscott-samos07.pdf, accessed: 2012/4/21.

[35] Shamir and Y. Tauman, "Improved Online/offline Signature Schemes," in *Proceeding 21st Annual International Cryptology Conference*, pp. 355-367, 2001.

[36] Raghupathy Sivakumar, Prasun Sinha, and Vaduvur Bharghavan, "Braving the Broadcast Storm: Infrastructural Support for Ad Hoc Routing," *Computer Networks*, vol. 41, no. 6, pp. 687-706, 2003.

[37] Y. Toor, P. Muhlethaler, and A. Laouiti, "Vehicle Ad Hoc Networks: Applications and Related Technical Issues," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 3, pp. 74-87, 2008.

[38] Huaqun Wanga and Yuqing Zhang, "On the Security of an Anonymous Batch Authenticated and Key Agreement Scheme for Value-Added Services in VANETs," *Procedia Engineering*, vol. 29, pp. 1735–1739, 2012.

[39] Tin-Yu Wu, S. Guizani, Wei-Tsong Lee, and Kuo-Hung Liao, "Improving RSU Service Time by Distributed Sorting Mechanism," *Ad Hoc Networks*, vol. 10, no. 2, pp. 212-221, 2012.

[40] Lo-Yao Yeh, Yu-Lun Huang, A.D. Joseph, S.W. Shieh, and W. Tsaur , "A Batch Authenticated and Key Agreement Framework for P2P-based Online Social Networks," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1907-1924, 2012.

[41] Chenxi Zhang, Pin-Han Ho, and Janos Tapolcai, "On Batch Verification with Group Testing for Vehicular Communications," *Wireless Networks*, vol. 17, no. 8, pp. 1851-1865, 2011.

[42] Chenxi Zhang, Xiaodong Lin, Rongxing Lu, Pin-Han Ho, and Xuemin Shen, "An Efficient Message Authentication Scheme for Vehicular Communications," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 6, pp. 3357-3368, 2008.

[43] Yun Zhou and Yuguang Fang, "Scalable and Deterministic Key Agreement for Large Scale Networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 12, pp. 4366-4373, 2007.