國立交通大學
資訊科學與工程研究所

博士論文

適用於車載網路中兼具安全與私密之機制研究

A Study on Security and Privacy Mechanisms for Vehicular Ad Hoc Networks

研究生：葉羅堯

指導教授：黃俊龍 博士

中華民國 九十九 年 十一 月

A Study on Security and Privacy Mechanisms for Vehicular Ad Hoc Networks

研究生：葉羅堯　　Student: Lo-Yao Yeh

指導教授：黃俊龍　　Advisor: Jiun-Long Huang

國立交通大學

資訊科學與工程研究所

博士論文

A Dissertation Submitted to

Institute of Computer Science and Engineering College of Computer Science

National Chiao-Tung University

In Partial Fulfillment of the Requirements

for the Degree of Doctor of

Philosophy in Computer Science

Hsinchu, Taiwan, Republic of China

November, 2010

適用於車載網路中兼具安全與私密之機制研究

學生：葉羅堯　　指導教授：黃俊龍 博士

國立交通大學 資訊科學與工程研究所

摘要

在未來我們可以期待，每台車輛上將安裝一個無線通訊設計，進而形成一個車載網路。而近年來，有眾多文獻正探討著車載網路上的安全與私密性議題。大多數的文獻圍繞在加強「安全相關應用」的訊息驗證。而在此博士論文中，我們首先將進一步利用車載網路特性來改進「道路救援效率」，我們提出一種以屬性為基礎的存取控制系統，簡稱為ABACS，來確保在緊急訊息上的安全性議題。藉由使用ABACS，最適合的緊急救援車輛將會被指派去執行救援的動作，並且此車輛將可安全地取得控制交通號誌的權利。ABACS 是基於一個新穎的密碼學技術，可達到訊息的機密性、合謀攻擊的預防性及細緻的存取控制。第二部分，我們則針對「加值型應用」的安全性。在此部分，針對改善安全性及可擴展性(scalability)，我們分別提出了「匿名式批次認證與金鑰協定(ABAKA)」及「具私密性的可攜式認證與存取控制協定(PAACP)」。為何需要可擴展性呢?主要是因為在車載網路中，每台車的速度介於每秒十公尺到四十公尺，相當於每小時三十六公里到一百四十四公里，因此有效率的認證機制將可有效地改善可擴展性。在ABAKA中，我們利用了批次認證的方式一次認證多個存取要求訊息，並且可同時建立起多把交談金鑰。而在PAACP中，我們利用了一個可攜式的證明文件來免除路側設備(RSUs)與服務伺服器(service provider)的長距離傳輸。透過分析與模擬的方式，我們可以證明這三個機制可有效地加強各種車載網路應用中的安全性、私密性與可擴充性。

關鍵字：屬性基礎加密法、存取控制、條件式隱私、身份證認、批次認證、金鑰協議。

A Study on Security and Privacy Mechanisms for Vehicular Ad Hoc Networks

Student: Lo-Yao Yeh Advisor: Dr. Jiun-Long Huang

Institute of Computer Science and Engineering College of Computer Science

National Chiao Tung University

Abstract

In the future, it is envisioned that each vehicle is equipped with a communication device to form a vehicular ad hoc networks (VANETs). Recently, several studies addressed security and privacy issues in VANETs. Most of them focused on authenticating traffic-related messages, one kind of safety applications. In this dissertation, we first aim to improve the efficiency of rescues mobilized via emergency communications over VANETs. An Attribute-Based Access Control System (ABACS) for emergency services with security assurance over Vehicular Ad Hoc Networks (VANETs) is proposed. With ABACS, the proper emergency vehicles are assigned to tackle the emergency event and delegated the authority to control traffic facilities. Using novel cryptographic preliminaries, ABACS realizes confidentiality of messages, prevention of collusion attacks, and fine-grained access control. Next, we embark on the security of value-added application. The Anonymous Batch Authenticated and Key Agreement (ABAKA) scheme and Portable privacy-preserving Authentication and Access Control Protocol (PAACP) are proposed to enhance the security and scalability issues for value-added applications in VANETs. In VANETs, the speed of a vehicle is changed from 10m/s to 40m/s (36km/hr to 144km/hr) and, therefore, the need of the efficient authentication is inevitable. ABAKA adopts the concept of batch verification to authenticate multiple requests sent from different vehicles and establish different session keys for different vehicles at the same time. PAACP adopts the concept of portable credentials to eliminate the backend communications with service providers. Throughout extensive analyses and simulation, we can show that these schemes can enhance the security, privacy and scalability issues for safety and non-safety applications over VANETs.

Keywords: Attribute-based encryption, access control, conditional privacy, authentication, batch verification, key agreement.

Acknowledgement

Special thanks go to my advisor Prof. Jinu-Long Huang and Yen-Cheng Chen for their guidance. Prof. Huang often teaches me the secret to doing research and gives me a lot of confidential to finish each work. Prof. Chen always carefully examines whether my work makes sense or not. Without them, this dissertation is never to be complete. Wish you can talk to your young students that you take pride in me as your student.

Next, I want to thank professors, including Ming-Syan Chen, Shi-Jinn Horng, Min-Shiang Hwang, John Kar-Kin Zao, Wen-Chih Peng, and Yu-Sung Wu, in my dissertation committee. They give a lot of valuable suggestions to improve this dissertation.

Then, I would also like to thank Prof. Shiuhpyng Shieh and Yu-Lun Huang for taking care me in the front part of my Ph. D student term. I am also grateful to have the chance to meet with Prof. Hung-Yu Chien. I learn a lot from his attitude in security field. In addition, Prof. Woei-Jiunn Tsaur usually broadens my horizen in the view of the education filed.

Finally, I want to dedicate this dissertation to my family. You are the hidden motivation to push me keeping harder and harder. I think, it is proud to you that our family has two Dr.s including a medical doctor and a philosophy doctor.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the future, it is envisioned that each vehicle is equipped with a wireless communication device, called on-board unit (OBU) to communicate with roadside units (RSU) located at street intersection. Such a network is called Vehicular Ad Hoc Network (VANET). The creation of VANETs is to improve the road safety. In VANETs, the communications protocol is based on DSRC protocol [1] and can be classified into: vehicle-to-roadside (RVC or V2I) and inter-vehicles (IVC or V2V). Figure 1.0.1 shows the network model of VANETs.

To improve road safety, vehicular ad hoc networks (VANETs) has been proposed. Several industries and academic have embarked on developing various applications in VANETs. In general, there are two kinds of applications, safety-related applications and value-added applications. The safety-related application aims to improve the road safety including emergency warnings, lane-changing assistance, intersection coordination, traffic sign violation warnings, and road-condition warnings [2]. In addition, the non-safety-related application (or called comfort application, value-added application) addresses on providing attractive commercial applications, such as Internet access, download maps, and multimedia files [3].

Before putting the above promising applications into practice in VANETs, several challenges are also emerged, such as security, privacy as well as scalability issues [2, 4]. The security issues includes bogus information attack, unauthorized preemption attack, message replay attack, message modification attack, impersonation attack, RSU replication attack, Denial-of-service (DoS) attack, and movement tracking [4]. On the other hand, some privacy requirements are essential for VANETs. A specified requirement is conditional privacy,

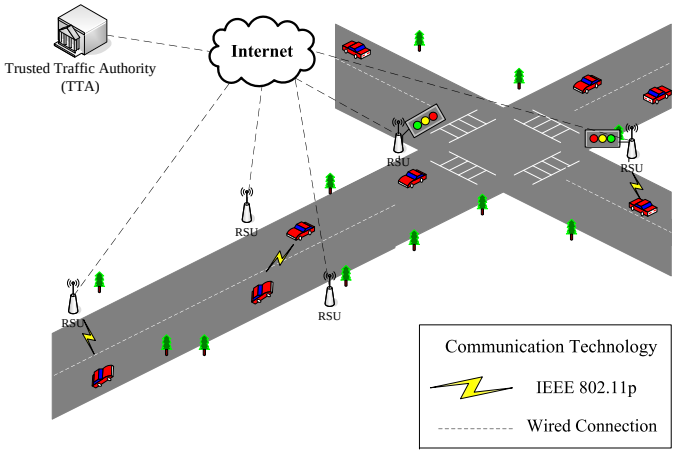


Figure 1.0.1: Network model of VANETs

which means user-related private information should be well-protected, while the traffic authorities (TA) gifts with the capability of reveal the identities of message senders in case of a traffic event dispute. As for scalability issue, the challenge is stringent time requirement [5, 6]. Since the speed of a vehicle can be up to 140 km/hr, the message response time should keep fewer than traditional wireless networks. The scale of VANETs should be regarded as very large because, according to DSRC protocol [1], each vehicle sends a safety-related message to RSU within a time interval of 100-300m ms. As a result, designing a security mechanism for vehicular network should take several aspects into account. Recently, several distinguish schemes [2, 4, 5, 6, 7, 8] have been proposed. Most of these schemes concentrated on improving the efficiency of message verification in the safety-related application. However, it is anticipated that the value-added applications [3] will gain more attention in the future. In this dissertation, we address on both applications as follows.

- For Safety-related Applications

Catering to the goal of VANETs, we aim to extend the merit of VANETs to facilitate the rescue efficiency. While an emergency event occurs, each vehicle will send an emergency event report to the adjacent RSU. We further take advantage of this emergency event report to dynamically assign the neighboring emergency vehicles (EVs), such as ambulances or police vehicles, to perform the rescue mission. Moreover, the assigned emergency vehicles are authorized to control the traffic facilities for accelerating the rescue efficiency. It is ob-

vious that the communication between Trusted Traffic Authority (TTA) and *EV*s should be well-protected against the eavesdropping of attackers. To achieve the goals, we proposed an attribute-based access control scheme (ABACS) for emergency service over VANETs. Extensive analysis and simulations are conducted to validate the proposed ABACS. In summary, the virtues of ABACS are including (1) better rescue efficiency, (2) better scalability, (3) secure communications and entity authentication.

- For Value-added Applications

1. We focus on designing the security mechanism for value-added applications in VANETs. As mentioned above, the scalability issue plays an important role in VANETs, which is also applied to value-added applications. With traditional PKI-based scheme, the service providers (SPs) are likely to be the bottlenecks because of the time-consuming verification. To mitigate this problem, we proposed an anonymous batch authentication and key agreement (ABAKA) scheme for value-added services over VANETs. By the performance evaluation, ABAKA outperforms the counterpart schemes and the ECDSA-based scheme adopted by IEEE Trial-Use standard [9] for VANET security. The merits of ABAKA scheme contain (1) mutual authentication, (2) session key agreement, (3) privacy preservation and conditional privacy, as well as (4) low transmission overhead and fast verification.

2. Next, we consider another type of value-added services over VANETs. In this work, we assume that RSUs are equipped with storage units to directly provide the location-related value-added service, such as local maps download. In traditional scheme [10], while a vehicle tries to access a service, the requesting vehicle sends a request message to an adjacent RSU, and the RSU has to pass the request message to the vehicle's SP for verifying the access admission and privileges of the requesting vehicle. However, the communication between the RSU and SP may cause a long delay up to 750-1200 ms [11]. Therefore, we propose a portable privacy-preserving authentication and access control protocol (PAACP) to reduce the communication between the RSU and SP. Furthermore, we strike on an attachable blind signature to achieve the sophisticated differentiated service access control instead of simple access admission. Through

theoretical analysis, PAACP can effectively reduce the waiting time for requesting a service. The advantages of PAACP include (1) mutual authentication, (2) session key establishment, (3) privacy preservation, (4) data confidentiality and integrity, (5) differentiated service access control, and (6) better scalability.

In the remainder of this dissertation is organized as follows. In Section 2, a brief overview of related work is presented. The first proposed scheme I- ABACS is described in Section 3. In Section 4, the second proposed scheme II- ABAKA is discussed, followed by the third proposed scheme III- PAACP is demonstrated in Section 5.

Chapter 2

Related Work

Several related studies [2, 4, 5, 6, 7, 8, 12] have been discussed on the security and privacy issues in VANETs. In this section, we simply classified these studies into several aspects with respect to the main goals of each scheme.

- Privacy Issue

1. To achieve the goals of message verification and anonymous, Raya *et al.* [12] first proposed a scheme where each vehicle is preloaded with a huge set of anonymous pairs of public keys and private keys. During a short period, a new anonymous public and private key pair is used to sign the next message. This scheme takes advantage of an assumption that vehicles are full of a large storage capability in VANETs. The traditional public-key-based cryptography is used as the security foundation. The main problem is the required large storage capacity to store the security information.

2. In the same year, Lin *et al.* [4] also proposed a scheme based on group signature for message verifications. Thanks to the properties of group signatures, the identity of a signer, or called a group member, will not be revealed except for the group manager. Therefore, the traffic authority can serve as the group manager. Consequently, the privacy problem can be well-protected and achieve the conditional privacy. Moreover, the extensive storage units are not required. The main weakness of this scheme is that the time for safety message verification grows linearly with the number of revoked vehicles in the revocation list.

3. To mitigate the problem in [4], an efficient conditional privacy preservation (ECPP) protocol [7] has been proposed. The main idea of ECPP is that each RSU will run a two-round protocol with each passing vehicle to issue a short-time anonymous certificate, which means each RSU takes on the role of the group manager. As a result, the linkability of the messages can be averted. An improvable point of ECPP is how to alleviate the dependence of RSUs while vehicles go into sporadic RSUs area such as rural area.

4. Similar to [7], Zhang *et al.* [2] also adopted a decentralized group authentication protocol to issue a on-the-fly signature for the passing vehicle. Moreover, the concept of batch verification and the situations of collapsed RSUs are taken into consideration. However, the problem of invalid requests in a batch does not be discussed in this paper. And, the problem of this scheme is the same as that of ECPP relying on RSUs to maintain the group signature, and RSUs are required to be trusted.

- Scalability issue

1. To cope with the scalability issue, an identity-based batch verification (IBV) scheme [5] has been proposed. According to DSRC protocol [1], each RSU has to verify 600 messages per second, while there are roughly 180 vehicles keeping within the communication range of an RSU. IBV scheme adopted the batch verification to improve the efficiency of message verifications. As a result, hundreds of messages can be verified in the same time. However, the batch verification goes to fail if there is a invalid messages in a batch.

2. In 2008, Zhang *et al.* [6] took the advantage of a key hash message authentication code (HMAC) to efficiently verify messages. The Diffie-Hellman exchange protocol is adopted to negotiate a symmetric key used in HMAC. A large number of anonymous certificates are also required to ensure the privacy issue. This scheme also relies on RSUs to broadcast the validity of messages transmitted by vehicles. As a result, if an RSU collapses, the scheme cannot enjoy the efficiency anymore.

3. Similar to [6], Lin *et al.* [8] employed the one-way hash chain and message authentication code (MAC) to verify messages. This scheme can work without the dependence on RSUs. However, a short period key disclosure delay is needed to avoid the abuse of hash chain value.

To sum up, most of existing security protocols [2, 4, 5, 6, 7, 8, 12] focused on safety-related applications to improve the efficiency of message verifications and to enhance the privacy preservation.

To the best of our knowledge, the emergency services in VANETs have yet discussed. As a result, we proposed ABACS scheme described in Section 3 to further exploit the properties of VANETs to enhance the rescue efficiency.

Moreover, the security problems in value-added applications have not been well-addressed inspiring us to investigate the value-added security mechanisms as shown in Section 4 and Section 5.

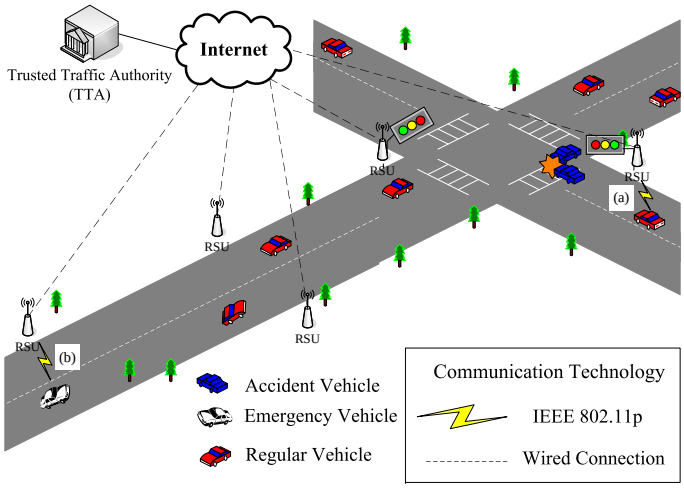
Chapter 3

ABACS: An Attribute-Based Access Control Scheme for Emergency Service

3.1 Motivation

Communications in VANETs can be classified into roadside-to-vehicle communication (RVC) and intervehicle communication (IVC). DSRC recommends that each vehicle should periodically broadcast traffic-related messages, including position information, current time, vehicle direction, speed, and acceleration/deceleration status. Furthermore, a vehicle will immediately transmit emergency messages when it witnesses a traffic accident. Thus, traffic jams or serious accidents can possibly be prevented if these traffic and emergency messages can be shared among vehicles. Essentially, the traffic-related messages are one-hop broadcasts without message relay, whereas emergency messages are transmitted in a multi-hop fashion to efficiently disseminate information about the occurrence of an emergency event.

Although many possible advantages of VANETs are known, some problems need to be overcome before VANETs can be employed widely. Recently, many studies [4, 5, 6, 8, 12, 13, 14, 15, 16] have addressed potential security and privacy issues in VANETs. Without security assurance in VANETs, any adversary can easily jeopardize a transportation system



(a) Vehicle detects the occurrence of an emergency event.
(b) TTA assigns appropriate emergency vehicles to deal with the emergency event and delegates the right to control traffic signals.

Figure 3.1.1: Emergency Event Processes

utilizing VANETs by disseminating bogus messages. Furthermore, vehicles involved in VANET communications may require privacy protection such that they cannot be tracked from the transmitted messages. Indeed, many solutions [2, 4, 5, 6, 7, 8] have been proposed to ensure the security and privacy of VANETs. However, most of these solutions focus on designing efficient and secure message authentication schemes for traffic-related messages. Some papers [17, 18, 19] address secure dissemination of emergency messages in the MAC layer. Only a few studies [13] have considered the security issues of emergency messages.

In this chapter, we discuss the secure utilization of VANETs to improve the rescue efficiency when an emergency event occurs. Because the introduction of VANETs is mainly driven by the need to enhance road safety, there is considerable demand for an effective communication process for dealing with a traffic emergency event. Instead of proposing an independent communication scheme for disseminating emergency messages, this chapter considers the entire rescue process for an emergency event as an emergency service. A typical scenario of an emergency service is illustrated in Figure 3.1.1. In this emergency scenario, after an emergency event occurs at a road intersection, a witness vehicle immediately reports the emergency event to Trusted Traffic Authority (TTA) through an adjacent RSU. TTA is responsible for assigning the most appropriate emergency vehicles (EVs), such as police vehicles and ambulances, to deal with the emergency event. Moreover, TTA

may delegate the authority of controlling traffic facilities, e.g., traffic signals in the neighborhood, to the assigned *EV*s for better rescue efficiency. During the emergency response, the communications between TTA and *EV*s should be well protected to ensure the security of message exchanges. The current IEEE Trial-Use standard [9] for VANET security adopts a traditional public-key-based signature scheme, ECDSA, for message authentications. The emergency service may involve many message encryptions and authentications, especially when TTA has to disseminate messages to many *EV*s with distinct public keys. As a result, the communications during the rescue process will be inefficient because several public-key-based encryptions are required for different *EV*s.

The *EV*s involved in an emergency service are usually those of certain types within a certain area, e.g., police vehicles in the neighborhood. Therefore, the abovementioned communications between TTA and *EV*s may be context-based. That is, TTA may broadcast a query message via the VANET to indicate the context of the emergency event, e.g., location, event type, or rescue requirements. Only *EV*s within the context will be notified to get involved in the emergency service. This chapter will make use of the context-based characteristic to develop a secure and efficient communication system. We introduce an **A**ttribute-**B**ased **A**ccess **C**ontrol **S**ystem for emergency services, named ABACS, over VANETs. To efficiently broadcast rescue-related messages to all *EV*s, ABACS exploits a novel fuzzy identity-based encryption [20] to realize secure one-to-many broadcast communications. In ABACS, each emergency vehicle is associated with a set of attributes, e.g., *State, County, District, Department, EV_type,* and *ELP* (Electronic License Plate)[12], where the *ELP* is used as the identification attribute of a vehicle. TTA will include a list of attribute values in a broadcast message based on the context of an emergency event. On receiving the broadcast message sent by TTA, each *EV* looks up the attributes and determines whether it is one of the *EV*s that the message is destined to. Moreover, only the *EV*s specified by the attributes can successfully decrypt the message. Accordingly, the most appropriate *EV*s will be selected to get involved in the rescue process. Therefore, the proposed ABACS affords the following advantages.

1. Rescue efficiency: According to the context of an emergency event, ABACS can ef-

fectively find the most appropriate EVs to handle the emergency event. For better rescue efficiency, these EVs also gain the authority to control traffic facilities from ABACS.

2. Scalability: Irrespective of the number of EVs selected, only one message will be broadcast by TTA. Furthermore, due to the nature of broadcasting, the message delivery does not require dynamic routing support in the VANET. Thus, ABACS achieves scalability in terms of the number of EVs.

3. Fine-grained access control: Using well-defined attributes, ABACS can enforce fine-grained access control among various types of EVs. When TTA broadcasts a rescue-related message[1] along with certain attributes, only those EVs that possess the selected attributes can access the rescue-related message.

4. Security properties: Message confidentiality and entity authentication can be realized in ABACS. With fuzzy identity-based encryption, rescue-related messages are well protected. Moreover, each assigned EV is implicitly authenticated by the attributes.

To the best of our knowledge, this work is the first study that addresses both the security and efficiency issues of emergency services in VANETs based on a provable cryptographic approach.

3.2 System Model and Cryptographic Preliminaries

3.2.1 System Model

A vehicular communication network for emergency services consists of two conceptual layers, as shown in Figure 3.1.1. The upper layer is composed of Trusted Traffic Authority (TTA) and RSUs. Connected with each RSU through a secure channel, e.g., the transport layer security (TLS) protocol, TTA is responsible for managing the overall traffic environment. Assume that, at critical intersections, RSUs are installed to serve as gateways to the lower layer. Some RSUs may be installed on traffic signal poles. The traffic signals can be

[1] In this chapter, the rescue-related messages include the Rescure Query Message (RQM), Rescure Response Message(RRM), and Mission Assign Messge(MAM) introduced in following section.

controlled via these RSUs. The lower layer is composed of regular vehicles and emergency vehicles (EVs), such as police vehicles, fire engines, and ambulances. In general, if there are EVs standby in emergency report centers (ERCs), these EVs can be easily notified to join a rescue mission via the fixed wired/wireless networks in ERCs. On the other hand, there are other EVs on patrol. ABACS can be used to effectively find patrolling EVs in the neighborhood and assign a rescue mission to near EVs for accelerating the rescue efficiency.

According to DSRC, the communication range of an RSU is typically larger than that of vehicles. We assume that TTA and RSUs trust each other and cannot be compromised by adversaries[2]. Moreover, TTA takes charge of public parameter settings and private value configurations for each EV.

3.2.2 Requirements

This chapter aims to develop a secure and efficient rescue process over VANETs. The functional requirements in terms of security and efficiency are presented as follows.

1. When receiving an emergency event report, TTA can secretly assign appropriate EVs to avert eavesdropping by malicious individuals or groups. Moreover, TTA can issue a traffic facility credential to the assigned EVs to control traffic facilities, such as traffic signals.

2. There are several kinds of emergency events whose rescues require EVs of different types. An efficient way to find desired EVs is essential to accelerate a rescue process.

3. It is possible that some EVs may be compromised by adversaries. The adversaries cannot benefit from the information held by the compromised EVs.

4. After an emergency event occurs, it is essential that a rescue mission could be enforced immediately and the rescue process be executed efficiently. An effective emergency

[2]Some schemes [2, 6] can be applied to implicitly authenticate RSUs to prevent the bogus RSU attack.

service should make use of VANET communications to achieve better rescue efficiency.

3.2.3 Design Objectives

To meet the above requirements, ABACS is proposed to achieve the following objectives.

1. Rescue-related message confidentiality. All rescue-related messages exchanged between TTA and *EV*s should be confidential without revealing any rescue-related information.

2. Fine-grained access control. Through fine-grained access control, only the desired *EV*s will be selected and authorized to join a rescue mission. Therefore, *EV*s can be recruited efficiently via VANETs.

3. Prevention of collusion attacks. If some *EV*s are compromised by an adversary, the adversary cannot combine parameters/attributes held by the compromised *EV*s to decrypt the rescue-related messages sent by TTA.

4. Rescue efficiency. TTA can communicate with *EV*s of certain types via a single encrypted rescue-related message sent over VANETs. The message can only be decrypted by specific *EV*s. As a result, the proposed scheme can efficiently find the most appropriate *EV*s and delegate the authority to control traffic facilities to them.

3.2.4 Cryptographic Preliminaries

Secret Sharing Scheme The concept of secret sharing was introduced by Shamir [21]. In a secret sharing scheme, a dealer distributes a secret s among a set of n players, $P = \{P_1, ..., P_n\}$. Each player P_i holds a piece s_i of the secret s. In order to recover the secret s, it is necessary to collect several or all pieces s_i of the secret s. A (t, n)-threshold secret sharing scheme is a particular case in which at least t pieces of s_i are required to retrieve the secret s. A typical secret sharing example is Shamir's threshold secret sharing scheme based on Lagrange polynomial interpolation [21], as described below.

Let Z_q be a finite field with $q > n$, and $s \in Z_q$ be the main secret to be shared. First, the dealer chooses a random polynomial $f(x)$ with degree $t-1$ such that $f(0) = s$. The polynomial can be written as $f(x) = a_0 + a_1 x + ... + a_{t-1} x^{t-1} = a_0 + \sum_{j=1}^{t-1} a_j x^j$ where $a_0 = s$ and $a_j \in_R Z_q$. Next, the dealer assigns a known value $\omega_i \in Z_q$ to each player P_i, and privately delivers the share $s_i = f(\omega_i)$ to P_i, for $i = 1,...,n$. As a result, a set of $\mathcal{L} \subset P$ with $|\mathcal{L}| \geq t$ is able to obtain the secret $s = f(0)$ by interpolating the set of shares s_i held by each $P_i \in \mathcal{L}$ as follows.

$$s = f(0) = \sum_{P_i \in \mathcal{L}} s_i \lambda_i^{\mathcal{L}} = \sum_{P_i \in \mathcal{L}} s_i \Big(\prod_{P_j \in (\mathcal{L} \setminus P_i),} \frac{x-j}{i-j} \Big)$$

where parameters $\lambda_i^{\mathcal{L}}$ are called the Lagrange coefficients. It has been proven that it is impossible to retrieve the secret s with less than t players [21].

3.3 Attribute-Based Access Control System (ABACS) for Emergency Services

In this section, we introduce the attribute-based access control system (ABACS) for emergency services in detail. Figure 3.3.1 illustrates the rescue process flow in the emergency scenario. A rescue process comprises an emergency event report phase, emergency vehicle recruiting phase, and rescue mission dispatch phase. In general, ABACS works as follows.

- Emergency event report phase: When an emergency event occurs, the witness vehicle sends an emergency event report message [13], which contains emergency event type and location, to an adjacent RSU. The RSU first confirms the validity of the emergency event report message.[3] If the emergency event report message is invalid, the RSU drops this message; otherwise, the RSU informs TTA of the emergency event.

- Emergency vehicle recruiting phase: After receiving the emergency event report from the RSU, TTA issues a rescue query message (RQM) to search the most appropriate EVs to deal with the emergency event. While obtaining an RQM, if an EV is

[3] Emergency event report messages can be verified by the current standard ECDSA method [9] or other schemes [13].

Table 3.1: Notations

Notation	Descriptions
EV	Emergency vehicle
RSU	Roadside unit
TTA	Trusted traffic authority
$\mathcal{UA}$	Universe attributes
$\mathcal{DA}$	Dummy attributes
$\widehat{ID}_{EV}$	Identity of an emergency vehicle
$\widehat{ID}_{M}$	Identity of message M
RQM	Rescue query message
RRM	Rescue response message
MAM	Mission assignment message
TFC	Traffic facility credential
G	Cyclic additive group
G_T	Cyclic multiplicative group
P	Generator of the cyclic group G
q	Order of the group G and G_T
$\hat{e}$	Bilinear map: $G \times G \rightarrow G_T$
d	Minimal number of overlapped attributes
$f(x)$	Polynomial with d-1 degrees
$\triangle_{i,S}$	Lagrange coefficient of a set S
t_i, y	Master keys of TTA, where $i = 1,...,\|\mathcal{UA}\| + d - 1$
z, v, r	Random numbers
σ	Credential signature
T_{expire}	Expired time for credential signature
$h(.)$	Collision-free one-way hash function such as SHA-1
$\|\|$	Message concatenation operation

available, the EV will send a rescue response message (RRM) back to TTA to confirm that it can tackle the emergency event.

- Rescue mission dispatch phase: Based on RRMs obtained from available EVs, TTA can determine which ones are most suitable for the rescue mission. Finally, TTA sends a mission assignment message (MAM), containing a traffic facility credential (TFC), to the assigned EVs. The TFC can be used to control the traffic facilities with the aid of RSUs for better rescue efficiency.

In ABACS, we focus on the design of the emergency vehicle recruiting phase and rescue mission dispatch phase, because the emergency event report phase can adopt the current standard ECDSA scheme or related works [13]. Note that the rescue-related messages, including RQM, RRM and MAM, should be well protected without leakage of information.

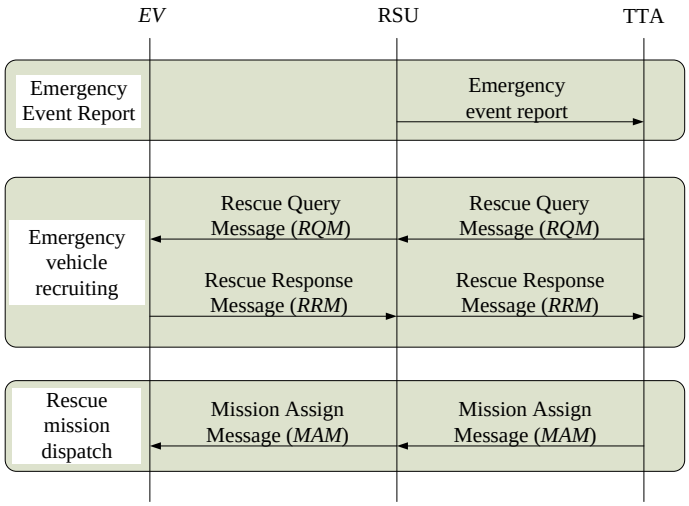


Figure 3.3.1: Rescue process flow for an emergency event

For ease of reference, Table 3.1 lists the notations used throughout the following description of the proposed system.

3.3.1 System Initiation

There exist various emergency vehicles. In ABACS, each EV can be described by a set of attributes. Let d be the minimal number of attributes required for selecting EVs by TTA to select EVs. In the following parameter setup phase, TTA will associate a random d-1 degree polynomial $f(x)$ with each EV with the restriction that the value of point 0 in each polynomial is the same, as denoted by $f(0) = y$.

3.3.1.1 Parameter Setup

Initially, TTA sets up the public parameters as follows. Let G be a cyclic additive group generated by P, and G_T be a cyclic multiplicative group. G and G_T have the same prime order q such that $|G| = |G_T| = q$. A security parameter k determines the size of the groups. There exists an admissible bilinear map $\hat{e}$: $G \times G \rightarrow G_T$ that satisfies the following properties.

1. Bilinearity: $\forall V,\ Q,\ R \in G$, and $\forall a, b \in Z_q^*$, $\hat{e}(Q, V + R) = \hat{e}(Q, V) \cdot \hat{e}(Q, R)$. In particular, $\hat{e}(aP, bP) = \hat{e}(aP, P)^b = \hat{e}(P, P)^{ab} = \hat{e}(P, aP)^b = \hat{e}(bP, aP)$.

2. Non-degenerate: If V, $R \in G$ then $\hat{e} : (V, R) \neq 1_{G_T}$.

3. Computability: There exists an efficient algorithm to compute $\hat{e}(V, R)$ for $\forall V, R \in G$.

The identity of each EV will be a subset of the universe attributes $\mathcal{UA}$. For instance, an EV can be identified by the following attributes {*State, County, District, Department, EV_type, ELP*} as identity $\widehat{ID}_{EV}$. In the list of attributes, EV_type is used to indicate the type of an EV, for instance, a police car or an ambulance. The *ELP* (Electronic License Plate)[12], i.e., car license number, can be independently used to uniquely identify an EV. When receiving a rescue-related message with identity $\widehat{ID}_M$, an EV can check whether $|\widehat{ID}_{EV} \bigcap \widehat{ID}_M| \geq d$. If yes, the EV can successfully decrypt the rescue-related message; otherwise, the rescue-related message is not meant for the EV and can be discarded.

According to the requirements of an emergency service, TTA first defines the universe attributes $\mathcal{UA}$. For simplicity, we assume 1,.., $|\mathcal{UA}|$-1 (mod q) are the indices used to represent all the possible universe attributes except for *ELP*. We use $|\mathcal{UA}|_{EV_i}$ to indicate the *ELP* attribute of each EV. Moreover, TTA also chooses d-1 dummy attributes $\mathcal{DA}$, which are used in mission assignments. Similarly, we assume ($|\mathcal{UA}|$ + 1),..., ($|\mathcal{UA}|$ + d - 1) as the indices required to represent all dummy attributes. Next, TTA chooses $t_1, ...t_{|\mathcal{UA}|-1}, t_{|\mathcal{UA}|_{EV_i}}, t_{|\mathcal{UA}|+1}, ..., t_{|\mathcal{UA}|+d-1}$ uniformly at random from Z_q, and selects y uniformly at random from Z_q. Finally, TTA publishes the following public parameters[4]:

$$T_1 = t_1 P,\ ...,\ T_{|\mathcal{UA}|-1} = t_{|\mathcal{UA}|-1} P,\ T_{|\mathcal{UA}|+1} = t_{|\mathcal{UA}|+1} P, ...,\ T_{|\mathcal{UA}|+d-1} = t_{|\mathcal{UA}|+d-1} P,\ Y = \hat{e}(P, P)^y.$$

The master key is:

$$t_1, ...t_{|\mathcal{UA}|-1}, t_{|\mathcal{UA}|_{EV_i}}, t_{|\mathcal{UA}|+1}, ..., t_{|\mathcal{UA}|+d-1}, y.$$

For ease of presentation, we define the Lagrange coefficient $\triangle_{i,S}$ for $i \in Z_q$ and a set S of attributes in Z_q:

[4]Note that the parameter $T_{|\mathcal{UA}|_{EV_i}}$ is only used for the *ELP* of an individual EV. It is not necessary to make the parameter public. However, the parameter could be queried from TTA while it is required in some cases.

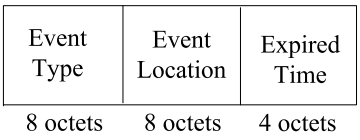


Figure 3.3.2: Rescue Query Message Format

$$\triangle_{i,S}(x) = \prod_{j \in S \setminus i} \frac{x-j}{i-j} \tag{3.3.1}$$

3.3.1.2 Key Generation

TTA is responsible for generating the private key values for each EV. First, it determines the proper attributes for describing an EV as its identity $\widehat{ID}_{EV} \subseteq \mathcal{UA}$, and randomly chooses a polynomial $f(i)$ with $d-1$ degree such that $f(0) = y$. Moreover, each EV also defines d - 1 dummy attributes $\mathcal{DA}$. The private key values for the EV are $(D_i)_{i \in \widehat{ID}_{EV} \cup \mathcal{DA}}$, where $D_i = \frac{f(i)}{t_i} P$ for each $i \in \widehat{ID}_{EV} \cup \mathcal{DA}$. These private key values are preloaded to the EV in the manufacture phase or via a secure channel outside the VANETs.

3.3.2 Emergency Vehicle Recruiting Phase

When receiving an emergency event report forwarded by any RSU, TTA will generate a rescue query message (RQM) and then broadcast it over the VANETs. RQM broadcasting is used to find appropriate EVs. If any EV is able to join the rescue mission, the EV will reply with a rescue response message (RRM). Both RQM and RRM should be transmitted securely. The proposed encryption scheme for RQM and RRM is described as follows.

1. Rescue Query Message (RQM)

   a) RQM Generation: TTA generates an RQM to query available EVs. The RQM includes the emergency event type, location, and expired time as illustrated in Figure 3.3.2.

   b) RQM Identity Selection: Before sending the RQM, TTA determines the identity $\widehat{ID}_{RQM}$ of the RQM based on the context of the emergency event. $\widehat{ID}_{RQM}$ is composed of a set of attribute values that describe the EVs required to join the

rescue mission. For instance, TTA may require EVs of certain types within a specific district or administrated by a certain department.

c) RQM Encryption: After generating $RQM \in G_T$ and selecting the proper identity $\widehat{ID}_{RQM}$, TTA chooses a random value $z \in Z_q^*$ that makes ABACS a probabilistic encryption scheme. Then, the encrypted RQM is published as

$$E = (\widehat{ID}_{RQM}, E' = RQM \cdot Y^z, \{E_i = zT_i\}_{i \in \widehat{ID}_{RQM}})$$

E is then broadcast over the VANETs.

d) RQM Decryption: When an EV receives an encrypted RQM, it determines whether the RQM can be decrypted by checking $|\widehat{ID}_{EV} \bigcap \widehat{ID}_{RQM}| \geq d$. If not, the encrypted message is discarded. If yes, the EV extracts theRQM by computing

$$E' / \prod_{i \in S} (\hat{e}(D_i, E_i))^{\triangle_{i,S}(0)} = RQM$$

where the attribute set $S = (\widehat{ID}_{EV} \bigcap \widehat{ID}_{RQM})$. The decryption can be verified as follows.

$$\begin{aligned}
& E' / \prod_{i \in S} (\hat{e}(D_i, E_i))^{\triangle_{i,S}(0)} \\
=& RQM \cdot \hat{e}(P,P)^{zy} / \prod_{i \in S} (\hat{e}(\frac{f(i)}{t_i} P, zt_i P))^{\triangle_{i,S}(0)} \\
=& RQM \cdot \hat{e}(P,P)^{zy} / \prod_{i \in S} (\hat{e}(P,P)^{zf(i)})^{\triangle_{i,S}(0)} \\
=& RQM \cdot \hat{e}(P,P)^{zy} / (\hat{e}(P,P)^{z(\sum_{i \in S} f(i) \cdot \prod_{j \in S \setminus i} \frac{x-j}{i-j})}) \\
=& RQM \cdot \hat{e}(P,P)^{zy} / (\hat{e}(P,P)^{z(y)}) \\
=& RQM
\end{aligned}$$

2. Rescue Response Message (RRM)

a) RRM Generation: The fields of an RRM are vehicle identity, vehicle type, vehicle location, vehicle direction, and vehicle velocity, as illustrated in Figure 3.3.3.

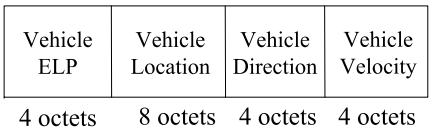


Figure 3.3.3: Rescue Response Message Format

b) RRM Encryption: To ensure the confidentiality, the EV chooses a random number $v \in Z_q^*$ and encrypts the RRM based on Elliptic Curve ElGamal encryption as follows.

$$C = (C'' = RRM + vT_1, V = vP)$$

C is then sent to TTA by a unicast over the VANETs.

c) RRM Decryption: While obtaining the ciphertext C, TTA extracts RRM by computing

$$\begin{aligned} & C'' - t_1 V \\ =& RRM + vT_1 - t_1 V \\ =& RRM + v \cdot t_1 P - t_1 \cdot vP \\ =& RRM \end{aligned}$$

3.3.3 Rescue Mission Dispatch Phase

After receiving RRMs in a predefined short time period, TTA will dispatch the most appropriate EVs to deal with the emergency event. TTA generates a mission assignment message (MAM) for the assigned EVs. For better rescue efficiency, the MAM contains a traffic facility credential (TFC) that is used to delegate the authority to control traffic facilities. Using the TFC, the assigned EVs can control traffic signals or other facilities around the area where an emergency event has occurred.

1. Mission Assignment Message (MAM)

a) MAM Generation: An MAM contains the traffic facility credential (TFC) and the credential signature (σ), as illustrated in Figure 3.3.4. Note that the TFC

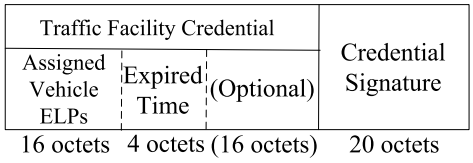


Figure 3.3.4: Mission Assignment Message Format

contains the ELPs[5] of the delegated EVs and T_{expire} for enabling the selected EVs to control traffic facilities before the time specified by T_{expire}. Note that if secure communications between the selected EVs are required, the optional field can be used for assigning a session key. To guarantee the validity of TFC , TTA also creates a credential signature σ as follows.

$$\sigma = h(TFC||T_{expire}) \cdot yP$$

a) MAM Identity Selection: TTA may assign multiple EVs to cooperatively tackle a serious emergency event. Therefore, TTA takes advantage of dummy attributes $\mathcal{DA}$ to ensure that only the assigned EVs can decrypt the encrypted MAM. TTA selects all the d-1 dummy attributes $\mathcal{DA}$ as well as the ELP attributes of the assigned EVs as the identity $\widehat{ID}_{MAM}$. Therefore, an assigned EV can decrypt the encrypted MAM based on its own ELP attribute as well as the d-1 dummy attributes.

b) MAM Encryption: In a manner similar to RQM encryption, TTA randomly selects $r \in Z_q^*$ and generates the ciphertext $\overline{E}$ as follows.

$$\overline{E} = (\widehat{ID}_{MAM}, \overline{E}' = MAM \cdot Y^r, \{E_i = rT_i\}_{i \in \widehat{ID}_{MAM}})$$

To avoid redundant transmissions, TTA only sends $\overline{E}$ by multicasting to those RSUs where the assigned EVs are converging.

c) MAM Decryption: While obtaining the ciphertext $\overline{E}$, each EV examines whether $|(\widehat{ID}_{EV} \cup \mathcal{DA}) \bigcap \widehat{ID}_{MAM}| \geq d$. If not, the EV drops the MAM; otherwise, the EV realizes that it has been assigned to go on the rescue

[5] In general, we assume that four EVs are assigned to handle an emergency event. In addition, the field of the assigned vehicle ELPs is extensible.

mission, and then obtains the TFC and σ[6] from MAM by calculating

$$\begin{aligned} & E'/\prod_{i\in S}(\hat{e}(D_i, E_i)^{\triangle_{i,S}(0)} \\ =& MAM \end{aligned}$$

where the attribute set $S = (\widehat{ID}_{EV} \cup \mathcal{DA}) \bigcap \widehat{ID}_{MAM}$. The verification of MAM decryption is shown as follows.

$$\begin{aligned} & E'/\prod_{i\in S}(\hat{e}(D_i, E_i)^{\triangle_{i,S}(0)} \\ =& MAM \cdot \hat{e}(P,P)^{ry}/\prod_{i\in S}(\hat{e}(\frac{f(i)}{t_i}P, rt_iP)^{\triangle_{i,S}(0)} \\ =& MAM \cdot \hat{e}(P,P)^{ry}/\prod_{i\in S}(\hat{e}(P,P)^{rf(i)})^{\triangle_{i,S}(0)} \\ =& MAM \cdot \hat{e}(P,P)^{ry}/(\hat{e}(P,P)^{r(\sum_{i\in S} f(i)\cdot \prod_{j\in S\setminus i} \frac{x-j}{i-j})}) \\ =& MAM \cdot \hat{e}(P,P)^{ry}/(\hat{e}(P,P)^{r(y)}) \\ =& MAM \end{aligned}$$

2. Traffic Facility Credential Verification

As a rescue mission proceeds, an assigned EV may send TFC and credential signature σ to ask RSUs to control traffic facilities. When an RSU receives the TFC and credential signature σ, the RSU believes the TFC is valid if $\hat{e}(\sigma, P) = Y^{h(TFC||T_{expire})}$, as verified below.

$$\begin{aligned} & \hat{e}(\sigma, P) \\ =& \hat{e}(h(TFC||T_{expire})yP, P) \\ =& \hat{e}(P,P)^{h(TFC||T_{expire})y} \\ =& Y^{h(TFC||T_{expire})} \end{aligned}$$

[6]The credential signature σ can be used to ensure the integrity of TFC and to implicily confirm that the MAM is sent by TTA.

If the verification is valid, the EV gains the authority to control the traffic signals/facilities governed by the RSU. After the EV applies the authority to control the traffic signal/facilities, the RSU will send a control acknowledgement to TTA to confirm that the EV has accepted the mission.

3.3.4 Discussion

3.3.4.1 Computational Delay

To further investigate the rescue efficiency, we first evaluate the computational delay of ABACS. In a manner similar to previous analyses [5, 7, 13], we mainly focus on the cost of point multiplication in elliptic curve and pairing computations, which require the most computation time. Let T_{mul} denote the time required to perform one point multiplication in an elliptic curve, and T_{pair} be the time required to execute a pairing operation. We adopt the experiment in [22] in which the processing time (in milliseconds) was observed for a super-singular curve of embedding degree $k = 6$ over $\mathbb{F}_{3^{97}}$ and executed it on an Athlon XP 2 GHz machine. The following results were obtained: $T_{mul} = 0.78$ ms and $T_{pair} = 2.82$ ms. Based on the computational delay of cryptographic operations, we can calculate the total computational delay of a complete round, denoted as T_{V_total}, in ABACS for an EV as follows.

$$\begin{aligned} T_{V\_total} &= (dT_{pair} + 1T_{mul}) + (2T_{mul}) + (dT_{mul} + 1T_{mul}) \\ &= 2dT_{pair} + 4T_{mul} \\ &= 2d \times 2.82 + 4 \times 0.78 \; ms \end{aligned}$$

That is, the decryption of RQM or MAM requires d T_{pair} for the product of sum and $1T_{mul}$ for point multiplication with Lagrange coefficient $\triangle_{i,S}$. The encryption of RRM requires 2 T_{mul} based on Elliptic Curve ElGamal cryptography.

3.3.4.2 Receiving Ratio Analysis

In the rescue mission dispatch phase of ABACS, TTA delivers an MAM to the assigned EVs. To minimize bandwidth consumptions, TTA only delivers the MAM by means of

multicasting to the RSUs whose signal coverage includes the assigned EVs. That is, the MAM is sent back to the same RSUs where the previous RRMs came from. However, EVs may move away from the RSUs during the emergency vehicle recruiting phase. Therefore, TTA has to generate the MAM within a stringent time limit. Moreover, to find the most appropriate EVs to deal with the emergency event, TTA needs to wait for a short time period ξ in order to receive more RRMs from different EVs. Therefore, we analyze the relationship between the vehicle movement speed v and the short waiting period ξ. The following assumptions are made to simulate a practical scenario:

- The average speed of emergency vehicles (denoted as v) ranges from 20 m/s ~ 50 m/s (or 72 km/hr ~ 180 km/hr). The valid coverage range of an RSU (denoted as C_{RSU}) is 300 m [8, 7].
- The number of attributes required for selecting an EV is 4 ($d = 4$) and 10 ($d = 10$). Therefore, the total computation delay T_{V_total} is $(2 \times 4) \times 2.82 + 4 \times 0.78 = 25.68$ ms and $(2 \times 10) \times 2.82 + 4 \times 0.78 = 59.52$ ms, respectively.

To evaluate the receiving ratio of an EV, we first estimate the required coverage range (denoted by C_{req}) over which an RSU successfully transmits the MAM to the assigned EV. The minimal required coverage range of an RSU is

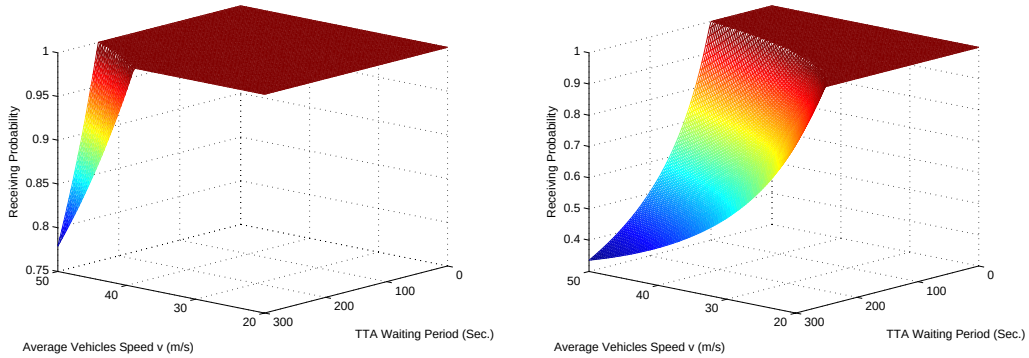
$$C_{req} = v \times T_{V\_total}$$

Then, we further discuss the receiving ratio, denoted as R_{ratio}, by considering the coverage range C_{RSU} of an RSU and the short waiting period ξ. The following formula can be used to estimate the receiving ratio R_{ratio}.

$$R_{ratio} = \frac{C_{RSU}}{C_{req} \times \xi} = \frac{C_{RSU}}{v \times T_{V\_totoal} \times \xi}$$

where $C_{RSU} \geq C_{req}$. Finally, R_{ratio} can be measured as

$$R_{ratio} = \begin{cases} 1 & \text{, if } \frac{C_{RSU}}{T_{V\_total}} \cdot \frac{1}{v \times \xi} \geq 1; \\ \frac{C_{RSU}}{T_{V\_total}} \cdot \frac{1}{v \times \xi} & \text{, otherwise.} \end{cases}$$



(a) Receiving ratio of an emergency vehicle ($d = 4$, $T_{V\_total} = 25.68$ ms)

(b) Receiving ratio of an emergency vehicle ($d = 10$, $T_{V\_total} = 59.52$ ms)

Figure 3.3.5: Receiving ratio of an emergency vehicle

Figure 3.3.5a, with $d = 4$ and $T_{V\_total} = 25.68$ ms, shows the receiving ratio with respect to velocity v, $20 \leq v \leq 50$, and waiting period ξ, $0 \leq \xi \leq 300$. It is observed that an EV can successfully receive the MAM when $20 \leq v \leq 39$ and $1 \leq \xi \leq 233$. Therefore, the proposed ABACS works well in most cases when $d = 4$. In the case of $d = 10$ and $T_{V\_total} = 59.52$ ms, the receiving ratio is shown in Figure 3.3.5b. The receiving ratio is 100% only when $1 \leq \xi \leq 100$. The analysis indicates that the receiving ratio decreases due to the greater computational delay that is caused by the use of a larger d. To cope with this problem, in the next subsection, a predictive transmission method is proposed to increase the receiving ratio.

3.3.4.3 Predictive Transmission

In this subsection, we propose a predictive transmission method to increase the receiving ratio for delivering MAMs. The following assumptions are required for the predictive transmission.

- TTA has the location information of each RSU, denoted as L_{RSU}.
- The transmission ranges of neighboring RSUs are partly overlapped.

The scenario of the predictive transmission is shown in Figure 3.3.6. In the emergency scenario, an EV has received an RQM via RSU1, and TTA performs predictive transmission

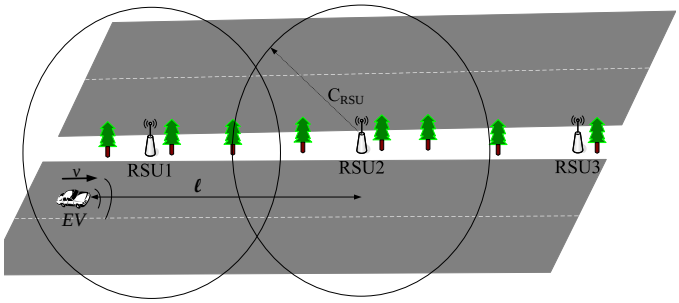


Figure 3.3.6: Predictive transmission scenario

for the neighboring RSU, i.e., RSU2, based on the probability that the EV enters the area covered by RSU2. Assume C_{RSU} is the transmission range of an RSU, Dir is the direction of the EV, and ℓ is the distance between EV and RSU2. Moreover, we assume that the locations of EVs are randomly distributed according to a uniform distribution, which has been widely assumed in previous literatures [8, 23, 24]. According to [8], the probability density function of the distance between the EV and reference RSU2 is measured as

$$f(\ell) = \frac{1}{v \cdot (\xi + T_{C\_TTA})}, 0 \leq \ell \leq v \cdot (\xi + T_{C\_TTA}) \tag{3.3.2}$$

where v is the velocity of the EV, ξ is the short waiting period, and T_{C_TTA} represents the delay caused by TTA in assigning the appropriate EV and generating the corresponding MAM. Based on the analysis of previous studies [8, 25], the velocity of an EV is assumed to follow a truncated Gaussian distribution with parameter $(\overline{v}, \sigma^2)$. Therefore, the probability (denoted as P_{enter}) that the EV enters the transmission range of RSU2 in the $(\xi + T_{C\_TTA})$ time interval can be measured as

$P_{enter} = P(\ell - C_{RSU} < v \cdot (\xi + T_{C\_TTA})|v)$

which can be expressed as follows.

$$\begin{cases} \text{if } Dir \text{ is } opposite, \quad P_{enter} = 0 \\ \text{if } Dir \text{ is } toward, \quad P_{enter=} \\ \iint P(\ell < v \cdot (\xi + T_{C\_TTA}) + C_{RSU} | v) f(\ell)\ d\ell dv, \\ = \frac{1}{\sigma\sqrt{2\pi}(v \cdot (\xi + T_{C\_TTA}))} \int_{vL}^{vH} (v \cdot (\xi + T_{C\_TTA}) + C_{RSU}) \\ \cdot exp(\frac{-1}{2}(\frac{v-\overline{v}}{\sigma})^2)\ dv \end{cases} \tag{3.3.3}$$

As a result, TTA can predict the entering probability P_{enter} to determine whether neighboring RSUs can help to deliver the MAM to the EVs.

3.4 Security Analysis

The security of the proposed ABACS is analyzed as follows.

1. Rescue-related message confidentiality: Based on Elliptic Curve Decisional Bilinear Diffie-Hellman (ECBDH) and Modified Bilinear Diffie-Hellman (ECMBDH) assumptions [20], the confidentiality of the rescue-related message is guaranteed. The security of the adopted fuzzy identity-based encryption has been proven in [20]. In [20], a fuzzy selective-ID model is used to show that the probability of the overall advantage of an adversary is only $\frac{1}{2}\epsilon$ for each bit. That is, an adversary cannot gain a advantage greater than guessing a bit without any information.

2. Fine-grained access control: In ABACS, each emergency vehicle (EV) possesses a set of attributes as its identity. Through the set of attributes, TTA can decide which types of EVs are able to decrypt the rescue-related messages to achieve fine-grained access control.[7] Because each EV holds a unique ELP and corresponding private key values, TTA can customize an identity for a multicast message intended to the desired EVs. Not only the computational delay but also the transmission overhead can be reduced by ABACS.

[7] To provide further fine-grained access control, the key-policy attribute-based encryption (KP-ABE) [26] can be adopted.

3. Prevention of collusion attacks: To prevent collusion attacks, ABACS randomly chooses different polynomials for distinct EVs. Therefore, each EV will keep different private key values generated with different polynomials. As a result, even if some EVs are compromised, an attacker cannot combine their private key values to derive the master private key values [20].

4. Rescue efficiency and security: One of the advantages of ABACS is the efficient communications between TTA and EVs, which are achieved by attributed-based multicast. Moreover, the assigned EVs can rapidly join a rescue mission with the aid of the received TFC for controlling traffic signals and facilities. As compared to the current VANET security standard [9], ABACS can secretly and efficiently deliver the rescue-related messages to EVs without requiring additional key-establishment phases [6].

3.5 Performance Evaluation

In this section, we evaluate the performance of ABACS in terms of computational delay and transmission overhead. To the best of our knowledge, there is no similar security scheme for emergency services. Therefore, we compared ABACS with the ECDSA scheme, which is adopted by the current IEEE1609.2 standard [9] as a security scheme for VANETs.

3.5.1 Computational Delay

As described in Section 3.3.4.1, the total computational delay (denoted as T_{V_total}) for an EV in ABACS is $2d \times T_{pair} + 4 \times T_{mul}$ ms. Here, we focus on the computational delay for TTA, because TTA is designed to handle all rescue-related messages sent from a number of EVs. Table 3.2 shows the computational delay of the dominant cryptographic operations, including point multiplication T_{mul} and bilinear pairing T_{pair}, for TTA in ABACS and ECDSA schemes in communications with a single EV or multiple EVs. Since ECDSA does not provide message confidentiality, we assume that the Elliptic Curve ElGamal encryption is adopted in ECDSA. Thus, it costs 2 T_{mul} for an encryption operation and 1 T_{mul} for a decryption operation. According to [5, 27], the time required to perform ECDSA signature

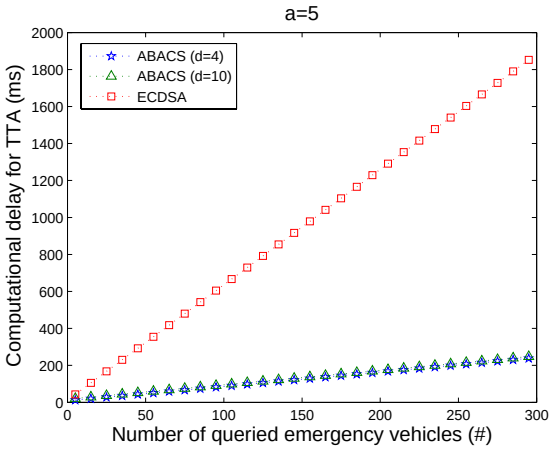
Table 3.2: Comparisons of computational delay for TTA (ms)

	Communication with a single emergency vehicle		Communication with n emergency vehicles	
	ABACS	ECDSA	ABACS	ECDSA
RQM Encryption	$(1+i)T_{mul}$	$3T_{mul}$	$(1+i)T_{mul}$	$3nT_{mul}$
RRM Decryption	$1T_{mul}$	$5T_{mul}$	nT_{mul}	$5nT_{mul}$
MAM Encryption	$(a+d-1)T_{mul}$	$3T_{mul}$	$(a+d-1)T_{mul}$	$3aT_{mul}$
Total	$(a+d+i+1)T_{mul}$	$11T_{mul}$	$(a+d+i+n)T_{mul}$	$(3a+8n)T_{mul}$

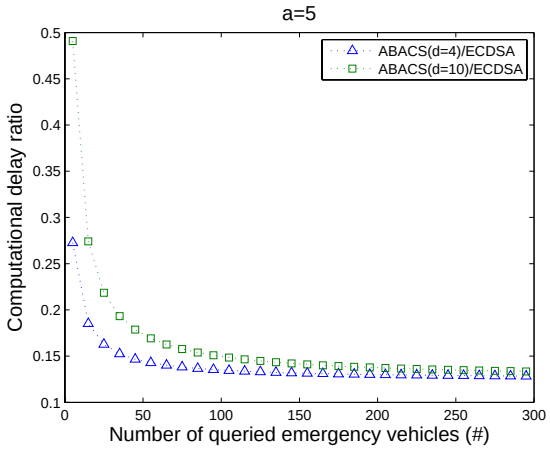
i: Total number of selected attributes ($i \geq d$), d: Minimal number of overlapped attributes; a: Number of the assigned EVs.

and certificate verification is 4 T_{mul}, and the time required to sign an ECDSA message is 1 T_{mul}. Therefore, an encryption in ECDSA costs 3 T_{mul} (EC ElGamal encryption + ECDSA signing) and a decryption in ECDSA costs 5 T_{mul} (ECDSA verification + EC ElGamal decryption). An RQM encryption in ABACS requires 1 T_{mul} for $\bar{E}$ and i T_{mul} for E_i, where i is the total number of selected attributes ($i \geq d$). For ease of evaluation, we set $i = d + 2$ in the following performance evaluation. A decryption in ABACS requires 1 T_{mul} for EC ElGamal decryption. The computational delay for an MAM encryption is $(a+d-1)T_{mul}$, in which aT_{mul} is for the ELPs attributes of the assigned EVs and $(d-1)T_{mul}$ is for the dummy attributes.

Referring to [22], the computational delay for T_{mul} and T_{pair} is 0.78 ms and 2.82 ms, respectively. Figure 3.5.1 (a) shows the relationship between the computational delay and the number of queried emergency vehicles (n), if the number of assigned EVs involved in a rescue is 5 ($a = 5$). It is observed that ABACS can greatly reduce the computational delays for different values of d. Moreover, Figure 3.5.1 (b) shows the ratio of the computational delay of ABACS to that of ECDSA. In a general rescue mission with only a few assigned EVs, ABACS is more than 80% faster than ECDSA when the number of queried EVs is greater than 40. Moreover, we investigate the computational delay for an disaster event requiring different numbers of assigned EVs. As shown in Figure 3.5.2 (a), when 100 EVs are queried, i.e., $n = 100$, ABACS achieves smaller computational delays than ECDSA for different numbers of assigned EVs. In fact, ABACS generates only an MAM for all the assigned EVs, whereas ECDSA has to produce distinct MAMs to individual EVs. This also explains why the computational delays in ABACS moderately increases by at most 163.8 ms ($d = 4$) and 173.16 ms ($d = 10$), as the number of assigned EVs increases. On the

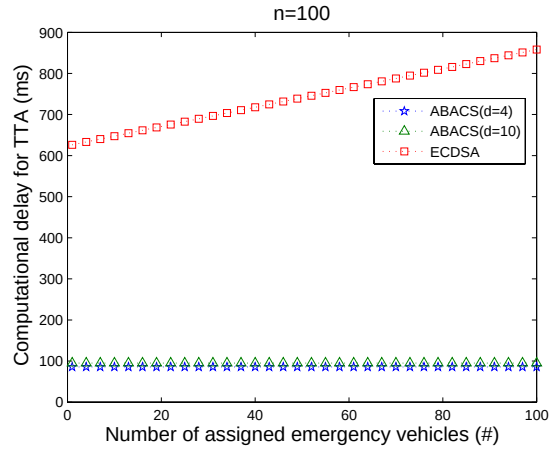


(a) Computation delay vs. number of queried emergency vehicles

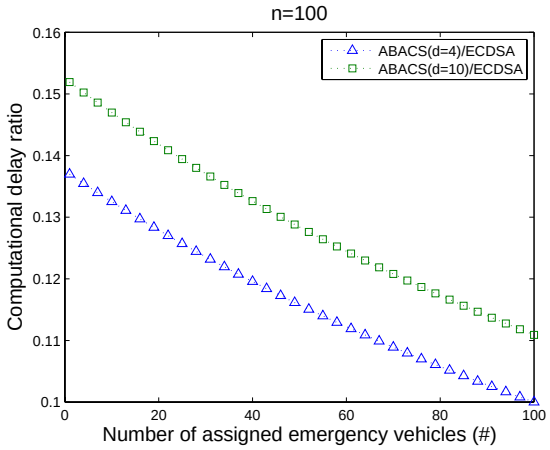


(b) Computational delay ratio vs. number of queried emergency vehicles

Figure 3.5.1: Computation delay evaluation in regular emergency events



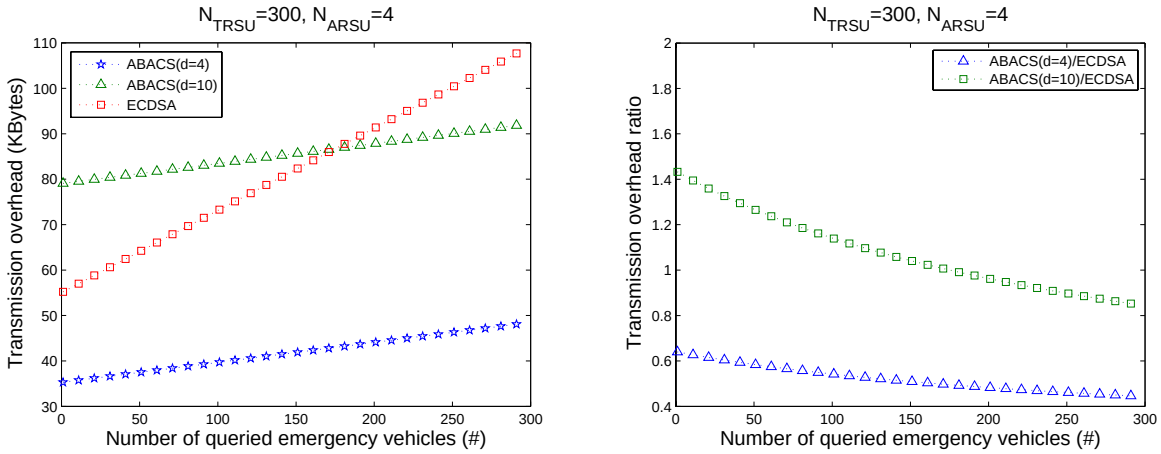
(a) Computational delay vs. number of assigned emergency vehicles



(b) Computational delay ratio vs. number of assigned emergency vehicles

Figure 3.5.2: Computational delay evaluation in disaster events

other hand, the computational delay in ECDSA also increases as more EVs are assigned; however, all are greater than 626.34 ms due to the computations in RQM and RRM. The computational delay ratios, illustrated in 3.5.2 (b), show that the computational delay of ABACS is only at most only 19% and 20.1% of that of ECDSA when $d = 4$ and $d = 10$, respectively.



(a) Transmission overhead vs. number of queried emergency vehicles

(b) Transmission overhead ratio vs. number of queried emergency vehicles

Figure 3.5.3: Transmission overhead evaluation

3.5.2 Transmission Overhead

In this section, we compare the transmission overhead of the two schemes. The transmission overhead mostly arises from the communications between TTA and the RSUs. The following evaluation focuses on the transmission overhead introduced by the signature, certificate, and encryption/decryption parameters, while the message itself is not considered. According to [9], the format of a signed message contains a 56-byte ECDSA signature and a 125-byte certificate. In ABACS, the transmitted parameters of RQM include 4*i bytes[8] for the identity and 20*i bytes for the decryption parameters, where i is the total number of selected attributes ($i \geq d$). As in Section 3.5.1, we set $i = d + 2$ in the following performance evaluation. As for RRM, the parameters consist of 4 bytes for the identity, and 20 bytes for the decryption parameters. With regard to MAM, the parameters consist of 4 * (a + d - 1) for the identity, 20 bytes for the encrypted credentials, and 20 * (a + d - 1) for the decryption parameters. According to DSRC [1], the bandwidth of a wireless data channel in VANETs is 10 MHz, corresponding to a channel data rate within the range of 3-27 Mb/s [28]. A typical data rate of 6 Mb/s is usually assumed for VANETs. Under the assumption of $d = 10$ and $i = 12$, the length of RQM will be 4*12 + 20 + 20*12 = 308 bytes. According to [5], there can be 180 vehicles within the communication range of an RSU. In a extreme

[8]We assume each attribute is of 4 bytes.

Table 3.3: Comparisons of transmission overhead (bytes)

	Communication with a single emergency vehicle	
	ABACS	ECDSA
RQM	$4i + 20i$	181
RRM	4 + 20	181
MAM	$4(a + d$ - 1) + 20 + $20(a + d$ - 1)	181
Total	$24(a + d + i$) + 20	543

Communication with n emergency vehicles		
	ABACS	ECDSA
RQM	$(4i + 20i)N_{TRSU}$	$181n \times N_{TRSU}$
RRM	$(4 + 20)n$	$181n$
MAM	$(24a + 24d$ - 4$)N_{ARSU}$	$181a \times N_{ARSU}$
Total	$(4i + 20i)N_{TRSU} + (24a + 24d$ - 4$)N_{ARSU} + 24n$	$181(n \times N_{TRSU} + a \times N_{ARSU} + n$)

N_{TRSU} : Total number of RSUs; N_{ARSU} : Number of RSUs where the assigned EVs are visiting.

case that all 180 vehicles are EVs, the demanded throughput for RQM is at most 0.42 Mb/s ($\frac{180 \times 1 \times 308 \times 8}{1024 \times 1024}$ Mb/s). Similarly, the throughput for RRM and MAM is 0.05 Mb/s and 0.45 Mb/s, respectively. Therefore, the maximal demanded throughput of ABACS is much smaller than 6 Mb/s.

Suppose that N_{TRSU} is the total number of RSUs and N_{ARSU} is the number of RSUs where the assigned EVs are visiting. Because the RQM is disseminated by broadcasting over N_{TRSU} RSUs, the transmission overhead of the RQM delivery can be estimated by $(4i + 20i)N_{TRSU}$. The transmission overhead of MAM is $(24a + 24d$ - 4$)N_{ARSU}$, because the MAM is only multicast to N_{ARSU} RSUs. Table 3.3 summarizes the transmission overhead in ABACS and ECDSA schemes. From Figure 3.5.3 (a), it can be seen that the transmission overhead of ECDSA is significantly higher than that of ABACS for $d = 4$ and $d = 10$. Because of the use of broadcasting and multicasting, the transmission overhead incurred by ABACS moderately increases as the number of queried EVs increases. Figure 3.5.3 (b) shows the ratio of the transmission overhead of ABACS to that of ECDSA is shown. It can be seen that the more the number of EVs are queried, the lower is the transmission overhead ratio that can be achieved. More precisely, when the number of queried EVs is greater than 61, the transmission overhead of ABACS for $d = 4$ and $d = 10$ is only 1.4% and 2.6% that of ECDSA, respectively.

Table 3.4: Simulation Parameters

City simulation area	1000m × 1000m
RSU Communication range	400 m
Simulation time	100 s
Wireless Protocol	802.11a
Wireless channel bandwidth	6 Mbs
Wired channel bandwidth	100 Mbs
Packet size for ECDSA message	181 bytes
Packet size for RQM message ($d = 4$ or 10)	164 or 308 bytes
Packet size for RRM message ($d = 4$ or 10)	40 bytes
Packet size for MAM message ($d = 4$ or 10)	184 or 328 bytes

3.5.3 Simulation

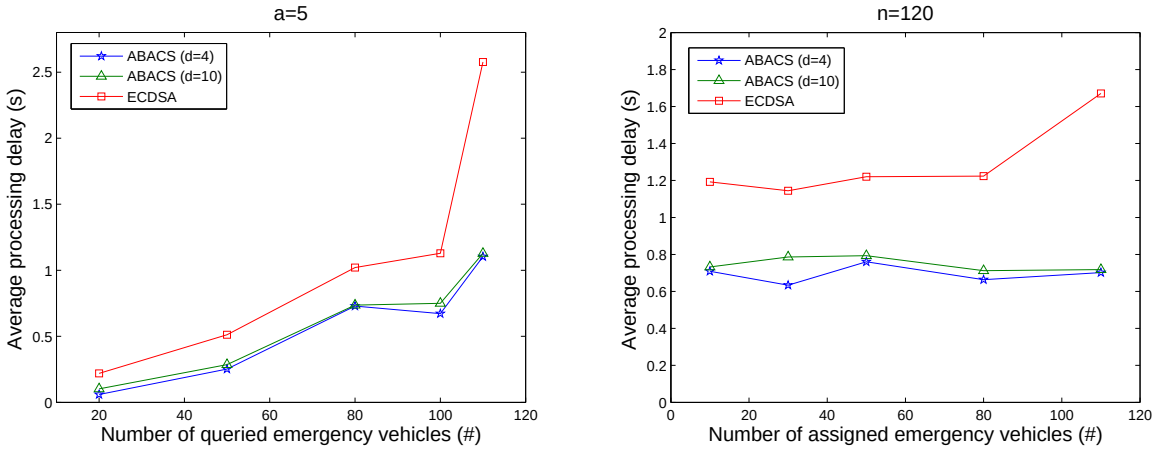
In addition to the theoretical analysis of computational delay in Section 3.5.1, we further evaluate the average processing delay and message loss ratio via a simulation on ns-2 [29]. In the simulation, we consider an area of 1×1 km^2 in urban areas. The simulation parameters are given in Table 3.4. We also adopt the TraNS [30] tool in the simulation for a better mobility model for vehicles. It is assumed that the maximum vehicle speed is 70 km/h. Predictive transmission is also implemented in the simulation. The Medium Access Control (MAC) protocol follows the IEEE 802.11a standard, which is the basis of DSRC [27, 30].

The average processing delay (denoted as $avgD$) is defined as

$$avgD = \frac{1}{N_r}\sum_{i=1}^{N_r}\frac{1}{N_{EV}}\sum_{j=1}^{N_{EV}}(T_{Recv}^{i,MAM,j} - T_{Send}^{i,RQM,j}),$$

where N_r is the number of emergency event reports, and N_{EV} is the number of EVs. $T_{Send}^{i,RQM,j}$ is the time when the application layer of TTA sends the rescue query message (RQM) of the i-th emergency event report to the j-th EV. $T_{Recv}^{i,MAM,j}$ is the time when the application layer of the $j-$th EV receives the mission assignment message (MAM) of the i-th emergency event report sent from TTA.

Figure 3.5.4a shows the average processing delay versus the number of queried EVs in regular emergency events. Note that the short waiting period (ξ) is not included in the average processing delay. As in Section 3.5.1, we assume the number of assigned EVs is 5 (a = 5). The simulation result shows that the average processing delay of ABACS (d = 4) is close to that of ABACS (d = 10), and ECDSA consumes more processing delay



(a) Average processing delay in a regular emergency event (b) Average processing delay in disaster events

Figure 3.5.4: Average processing delay

than the others. It is also seen that the more EVs are queried, the more advantages of ABACS can be achieved. This result is basically the same as the analysis shown in Figure 3.5.1a. The simulation result for disaster emergency events is shown in Figure 3.5.4b. In general, there are only slight variations of processing delay in ABACS with respect to the number of assigned EVs. However, the processing delay of ECDSA increases as more EVs are involved. This result also corresponds with the analysis shown in Figure 3.5.2a.

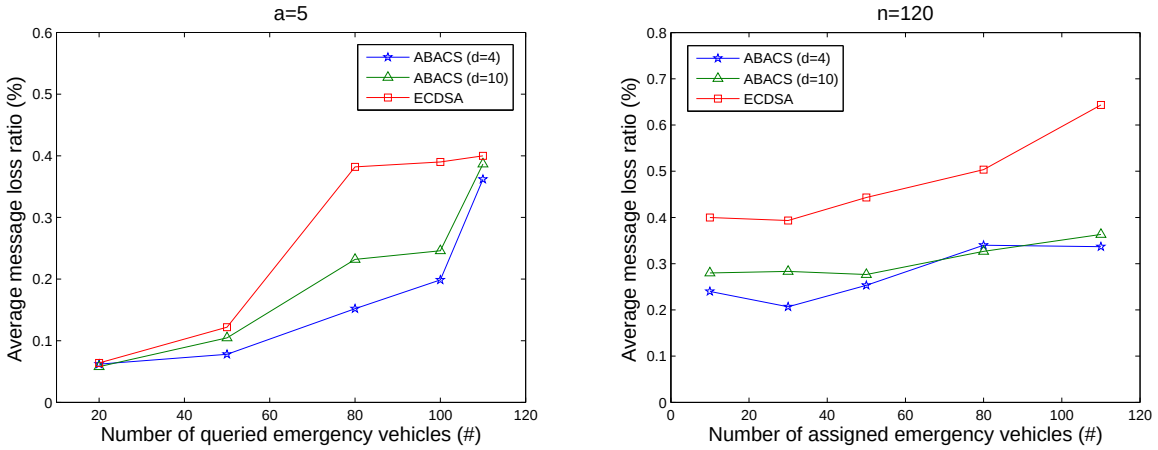
The average loss ratio, denoted as $avgLR$, is defined as

$$avgLR = \frac{1}{N_r}\sum_{i=1}^{N_r}\frac{1}{N_{EV}}\sum_{j=1}^{N_{EV}}\left(\frac{M_{Recv}^{i,RQM,j}+M_{Recv}^{j,RRM,i}+M_{Recv}^{i,MAM,j}}{M_{Send}^{i,RQM,j}+M_{Send}^{j,RRM,i}+M_{Send}^{i,MAM,j}}\right),$$

where N_r is the number of emergency event reports. $M_{Send}^{i,RQM,j}$ is the number of RQMs sent to the j-th EV for the i-th emergency event report[9], $M_{Send}^{j,RRM,i}$ is the number of RRMs sent by the j-th EV for the i-th emergency event report, and $M_{Send}^{i,MAM,j}$ is the number of MAMs sent to the j-th EV for the i-th emergency event report. $M_{Recv}^{i,RQM,j}$ represents the number of RQMs received by the j-th EV for the i-th emergency event report, $M_{Recv}^{j,RRM,i}$ represents the number of RRMs received by TTA for the i-th emergency event report, and $M_{Recv}^{i,MAM,j}$ represents the number of MAMs received by the j-th EV for the i-th emergency event report.

Figure 3.5.5a shows the relationship between the average loss ratio and the number of

[9]Note that messages sent via broadcasting should be counted multiple times as many as the number of receivers.



(a) Average message loss ratio in regular emergency events (b) Average message loss ratio in disaster events

Figure 3.5.5: Average loss ratio

queried EVs in regular emergency events. The loss ratio of ECDSA is up to about 40% when the number of queried EVs is more than 50, while ABACS attains the same loss ratio when the number of queried EVs is more than 100. Furthermore, we also investigate the loss ratio in disaster events, shown in Figure 3.5.5b. It can be seen that the loss ratio of ECDSA rapidly increases as the number of assigned EVs grows. The reason is that in ECDSA there needs a dedicated MAM message for each assigned EV. Each MAM message is encrypted using the public key of the EV, and is sent separately over the VANET. On the other hand, the loss ratio of ABACS only gradually rises no more than 40% as the number of assigned EVs increases, because in ABACS only one encrypted MAM message is required. It can be observed that the average loss ratio of the two ABACS-based schemes is only slightly affected by the number of assigned EVs. Some studies [17, 18] in the MAC layer can be used to further improve the packet loss problem.

Chapter 4

ABAKA: An Anonymous Batch Authentication and Key Agreement Scheme for Value-Added Service

4.1 Motivation

The creation of VANETs is to enhance the road safety and improve drivers' driving experiences, which is called safety-related applications. Onboard units (OBUs) equipped in vehicles periodically broadcast routine traffic-related messages with information like the position, current time, direction, speed, acceleration/deceleration, and traffic events, etc. By that information, driver can catch on a better understanding of their driving environment. In addition, value-added applications, or called non-safety applications, can also be envisioned to offer various entertaining services to drivers and passengers. The convenient value-added services include Internet access, navigation, instant messenger, toll payment service, and electronic advertisements [12, 31].

Along with the growth of VANETs, several challenges are emerging such as security and privacy issues. Prior to realizing the enjoyable value-added applications into practice in VANETs, we have to deal with the security and privacy issues [5, 6, 7, 8, 10]. Fundamentally, we must guarantee the identity authentication and data integrity. In value-added

applications, the confidentiality is also required. In addition, the requirement of privacy preservation must be reached in the sense that user-related private information, including user identity, and user location etc.. Although several studies [4, 5, 6, 10, 12, 13, 32] have addressed aforementioned issues, most of them are designed for safety-related applications to ensure the message verification and integrity. It is obvious that attractive value-add services play an important role to raise the interests of the consumers to take in VANETs. On the other side, due to the speed of vehicles varying from 36 km/hr to 140 km/hr [9], there is a unique stringent time requirement in the vehicular communication [5, 10]. According to the DSRC standard [1], a vehicle sends a safety-related message to its neighboring RSU every 100-300 ms, which means an RSU has to verify some 600 safety-related messages per second if there are roughly 180 vehicles keeping within the communication range of the RSU [5]. In other words, the security scheme for value-added applications should not pose a heavy burden on RSUs. Therefore, the burden may gather at a single authentication server which incurs the bottleneck problem. Obviously, it is critical to develop an efficient and secure authentication scheme before value-added applications can come into effect.

In order to tackle the above mentioned problems including security, efficiency and scalability problems, we proposed an Anonymous Batch Authentication and Key Agreement scheme, named ABAKA, to build a secure environment for value-added services in VANETs. To avoid bottleneck problems, ABAKA is inspired by the concept of the batch verification [5] to simultaneously authenticate multiple requests sent from different vehicles using elliptic curve cryptography (ECC), which is adopted by the IEEE Trial-Use standard [9]. Meanwhile, multiple session keys for different vehicles can also be negotiated in the same time. To the best of our knowledge, this is the first study that provides batch authenticated and key agreement for value-added applications in VANETs. ABAKA enjoys the following unparalleled features: 1) Multiple vehicles can be authenticated at the same time rather than one after the other. It is an appealing solution to elaborately solve the possible bottleneck problems; 2) Not only batch authentication can be achieved but batch key agreement can be accomplished as well. Depended on different key agreement parameters sent from the requesting vehicles, ABAKA could negotiate distinct session key with each vehicle to ensure the confidentiality of subsequent messages; 3) By creating distinct pseudo identities and the

corresponding private keys, the privacy regarding the real identity of vehicle and private information is guaranteed; 4) The real identities of the vehicles can be uniquely revealed by the service provider under specific conditions; 5) Thanks to the advantage of tamper-proof devices in vehicles, the efforts on the storage cost and the transmission overhead can be significantly alleviated.

4.2 System Model and Preliminaries

4.2.1 System Model

We introduce a two-layer vehicular network model for value-added applications, as shown in Figure 4.2.1. The lower layer is composed of vehicles and RSUs. The communications, either Inter-vehicle (IVC) or roadside-to-vehicle (RVC), are based on Dedicated Short Range Communications (DSRC) standard [1]. According to DSRC standard, the communication range of an RSU is adjustable, so it can be larger than that of the vehicles, meaning that some vehicles can hear messages sent from the RSU while the RSU may not hear messages sent from the vehicles. The upper layer is comprised of various service providers (SPs) and a Trust Authority (TA). The service providers have made a contract with the TA because the SPs will setup the system parameters in vehicles by the aid of the TA. The SPs can be connected with RSUs through secure channels, such as transport layer security (TLS) protocol by wired or wireless connections. The SPs provide various services, such as multimedia streams, instant messenger, and navigation services etc., and RSUs serve as gateways to deliver data to the requesting vehicles. According to the current VANET security standard [9], before messages are sent, OBUs should sign the messages with their private keys issued by TA to ensure the integrity of messages. Then, each RSU is responsible for checking the integrity and forwarding the valid messages to the SPs.

In this chapter, ABAKA aims at addressing the security between the SPs and vehicles. We assume that TA are trusted and will never be compromised, which is often assumed in VANETs schemes [5, 6, 7]. And, the SPs will not be compromised in the system initialization phase.[1]

[1]The TA can adopt Kerberos scheme [33] to guarantee the genuineness of the SPs in the system initial-

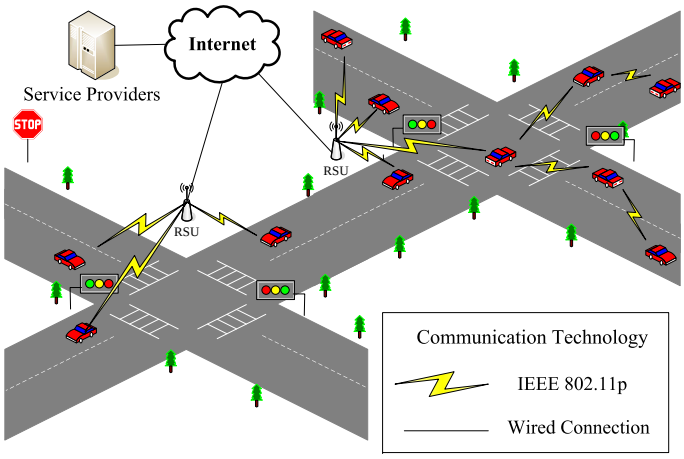


Figure 4.2.1: The network model for value-added service

4.2.2 ECC Preliminaries

An elliptic curve is a cubic equation of the form $y^2 + axy + by = x^3 + cx^2 + dx + e$, where a, b, c, d and e are real numbers. In an elliptic curve cryptosystem (ECC), the elliptic curve equation is defined as the form of $E_p(a, b)$: $y^2 = x^3 + ax + b \pmod{p}$ over a prime finite field F_p, where $a, b \in F_p$, $p > 3$, and $4a^3 + 27b^2 \neq 0 \pmod{p}$ [34]. Given an integer $s \in F_p^*$ and a point $P \in E_p(a, b)$, the point-multiplication sP over $E_p(a, b)$ can be defined as $sP = \underbrace{P + P + \cdots + P}_{s}$. In general, the security of ECC depends on the difficulties of the following problems [35, 36].

Definition 1: Given two points P and Q over $E_p(a, b)$, the elliptic curve discrete logarithm problem (ECDLP) is to find an integer $s \in F_p^*$ such that $Q = sP$.

Definition 2: Given three points P, sP, tP over $E_p(a, b)$ for $s, t \in F_p^*$, the computational Diffie-Hellman problem (CDHP) is to find the point $(s \cdot t)P$ over $E_p(a, b)$.

Up to now, there is no polynomial algorithm enable to solve any of the above problems [35, 36]. As compared to the counterpart scheme [5], the proposed scheme exploits the point multiplication over ECC instead of bilinear pairing to reduce the computational cost.

ization phase.

4.2.3 Security Objectives

For value-added applications in VANETs, a secure system should meet the following security objectives.

1. Mutual authentication: The communication parties should be authenticated to guard against the impersonation attack. The previous prominent works [4, 5, 6, 7, 8, 13] are designed for safety-related applications focusing on the message authentication. For value-added applications, the mutual authentication between the vehicles and service providers is essential.

2. Session key establishment: To ensure the data confidentiality, session key establishment is indispensable for value-added applications in VANETs. With the session key, the service provider can build a secure communication path with the requesting vehicle for subsequent communications for various value-added services such as multimedia streams.

3. Privacy Preservation: The identities of vehicles should be hidden from a message receiver during the authentication process for keeping the senders' personal information private.

4. Low transmission overhead and fast verification: Due to the stringent time requirement in VANETs, the security scheme should consider the efficiency into account. The transmission overhead should be lower the better, and a bunch of requests should be verified as soon as possible.

4.3 Anonymous Batch Authenticated and Key Agreement Scheme (ABAKA)

In this section, we propose a novel Anonymous Batch Authenticated and Key Agreement scheme (ABAKA) for value-added applications in VANETs. ABAKA consists of the following three phases: the system initiation phase, the pseudo identity generation phase, and the batch authentication and key agreement phase. A new vehicle first performs the system

Table 4.1: Notations

Notation	Descriptions
V_i	The ith vehicle
RSU	A roadside unit
SP	A service provider
$RVID$	The real identity of a vehicle
SID	The identity of the service provider
G	A cyclic additive group
P	The generator of the cyclic group G
q	The order of the group G
v	The private secret of the tamper-proof device
PK_{SP}	The SP's public key preloaded in each vehicle
RK_{SP}	The SP's private key
PWD	A password to activate a tamper-proof device
ID_i	A pseudo identity of the vehicle V_i
ID_i^j	A part of the ID_i, where j=1 or 2 and ID_i=(ID_i^1|| ID_i^2)
mrk	A master private key
CRK_i	A corresponding private key of the vehicle V_i
T	The timestamp
$\triangle T$	The predefined endurable transmission delay
KP	The key parameter used for key agreement
$h(.)$	A collision-free one-way hash function such as SHA-1
$H(.)$	A MapToPoint hash function such as H: $\{0, 1\}^* \rightarrow$G
||	Message concatenation operation
$\oplus$	Exclusive-OR operation

initiation phase to preload the system parameters. Then, the pseudo identity generation phase is used to generate the pseudo identity and corresponding private key for privacy issue. Finally, the batch authentication and key agreement phase is executed when the vehicle wants to access services provided by SPs. The notations throughout this chapter are listed in Table 4.1.

4.3.1 System Initiation

First, we assume each vehicle is equipped with a tamper-proof device, which is secure against any compromise attempt in any circumstance. Note that the use of a tamper-proof device is recommended by the current VANET security standard [9] and several famous VANET schemes [5, 12] to reduce the risk of vehicles compromised by adversaries. Thanks to tamper-proof devices on vehicles, an adversary cannot attain any data stored in the device [5, 12, 32]. Initially, the SP sets up the following system parameters for vehicles that

have made purchase contracts with the SP.

- Let G be a cyclic additive group generated by P with the order q.
- The SP randomly chooses $v \in \mathbb{Z}_q^*$ as the private secret, and v will be loaded in the vehicles' tamper-proof devices.
- Each vehicle is preloaded with the public parameters $\{G,\ q,\ P,\ PK_{SP},\ h(),\ H()\}$, and $\{v\}$ is preloaded in the tamper-proof device. After receiving $\{v\}$, each vehicle computes the master private key $mrk = h(v||SID)$ to store in the tamper-proof device as well.

4.3.2 Pseudo Identity Generation

For privacy preservation, similar to [5], the tamper-proof device is responsible for generating random pseudo identities and corresponding private keys (CRK) based on elliptic curve cryptography (ECC) [34]. The tamper-proof device consists of three modules: the authentication module, pseudo identity generation module, and corresponding private key generation module. Figure 4.3.1 shows the procedures of tamper-proof device.

Authentication module: The authentication module is used to ensure the validity of the user. A user inputs its unique real vehicle identity $RVID_i \in G$ and $PWD_i \in \{0, 1\}^*$ to pass the verification of the authentication module. If both $RVID_i$ and corresponding PWD_i are valid, $RVID_i$ is delivered to the next module, pseudo identity generation module; otherwise, the tamper-proof device refuses to activate itself. Ensured by the authentication module, an adversary cannot get any information even though the tamper-proof device is compromised by the adversary.

Pseudo identity generation module: The pseudo identity generation module takes charge of generating a random pseudo identity ID_i for the purpose of anonymity. Each ID_i is composed of two parts ID_i^1 and ID_i^2. Upon receiving $RVID_i$, the pseudo identity generation module chooses a random number $w_i \in \mathbb{Z}_q^*$ to create a point $R_i \in G$ such that $R_i = (x_i,\ y_i) = w_i P$. Then, let $ID_i^1 = h(R_i)$ and $ID_i^2 = RVID_i \oplus H(v\|ID_i^1)$, which allows only the SP to reveal the real identity $RVID_i$ of V_i. Finally, the tamper-proof device sends ID_i to next module, private key generation module.

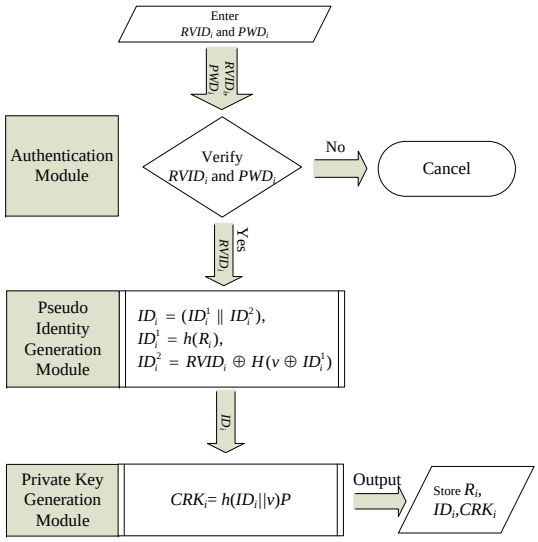


Figure 4.3.1: The procedures of tamper-proof device

Private key generation module: This module manages the generation of the corresponding private key based on the pseudo identity ID_i. The corresponding private key CRK_i is set to $h(ID_i||v)P$.

In the end, V_i can store a list of random point R_i, and pseudo identities $ID_i = (ID_i^1||ID_i^2)$ with its corresponding private keys CRK_i. Note that the generation of pseudo identities and private keys can be finished offline without any delay.

4.3.3 Batch Authentication and Key Agreement

In this phase, there are three kinds of procedures: the request procedures, vehicle verification and key agreement procedures, as well as mutual authentication and key agreement procedures. Firstly, the request procedures are initiated by a vehicle when the vehicle wants to access some services provided by a SP. Next, the vehicle verification and key agreement procedures are performed by the SP to check the validity of the requesting vehicle and to negotiate a session key for the confidentiality of subsequent communications. Lastly, the mutual authentication and key agreement procedures are executed by the requesting vehicle to check the validity of the SP. After performing the three procedures, a session key shared by both the requesting vehicles and the SP is generated to secure the subsequent communications.

Pseudo Identity (ID)	Material Message (M)	Verification Message (F)	Timestamp (T)
40 bytes	20 bytes	20 bytes	4 bytes

Figure 4.3.2: The request packet format

We first introduce the request procedures launched by vehicles. Next, we elaborate on the vehicle verification and key agreement procedures in term of dealing with a single request and multiple requests. Finally, the mutual authentication and key agreement procedures are discussed to ensure the validity of the SP.

4.3.3.1 *Request Procedures*

With the tamper-proof device, V_i obtains a random point $R_i \in G$, a pseudo identity ID_i, and the corresponding private key CRK_i as well as the master private key mrk. To issue a request, V_i executes the following procedures:

1. To ensure the freshness, V_i first generates $t_i = h(T_i)$, where T_i denotes the current timestamp. Note that it is assumed that each vehicle can perform the time synchronization by tamper-proof device [12].

2. With the above values, V_i can calculate $M_i = R_i + t_i CRK_i$, and $F_i = (mrk \cdot x_i)P$, where x_i is the x coordinate of point R_i.

3. Finally, according to the request packet format shown in Figure 4.3.2 , V_i delivers the request packet $<ID_i, M_i, F_i, T_i>$ to the SP by the aid of the neighboring RSU.

Notice that although there is no explicit key agreement parameters transmitted in the request packet, ABAKA takes the advantage of the random point R_i as the Diffie-Hellman key agreement parameter for saving the transmission overhead. The request message packet defined in Figure 4.3.2 is composed as follows. The pseudo identity ID is in the first field; the second field is the material message M; the verification message F is in the third field; the last field stores the current timestamp T for withstanding replay attacks. Here, ABAKA adopts SHA-1 as the underlying hash algorithm, and uses the MNT curve [37] with 160-bit prime order q.

4.3.3.2 *Vehicle Verification and Key Agreement Procedures*

Based on the system model described in Section 3.2, the SP is responsible of authenticating and negotiating a session key with each requesting vehicle. In some situations, numerous requests may crowd in the SP at the same period. To mitigate the possible bottleneck problems, we propose a batch verification and key agreement scheme. For the ease of presentation, we introduce the verification of a single request, followed by the presentation on the batch verification of multiple requests.

***Single request authentication*:** Given the request $<ID_i, M_i, F_i, T_i>$ from V_i, the SP performs the steps as follows.

1. For freshness, we assume the receiving time is T_{SP_now}. The SP checks whether $\triangle T \geq T_{SP\_now} - T_i$ is valid, where $\triangle T$ is the predefined endurable transmission delay. If yes, then go to step 2; otherwise, the SP ceases this connection.

2. To ensure the legitimacy of this request, the SP calculates $CRK_i = h(ID_i \| v)P$ depended on the pseudo identity ID_i, public parameters as well as his own private secret v, and computes $t_i = h(T_i)$. With the CRK_i and t_i, the SP acquires the point $\hat{R}_i = M_i - t_i CRK_i$, where $\hat{R}_i = (\hat{x}_i, \hat{y}_i)$ and verifies whether $F_i \stackrel{?}{=} (h(v||SID) \cdot \hat{x}_i)P$ is held or not. If so, then go to next step; otherwise, this connection is terminated.

3. For mutual authentication, the SP picks a random number $z \in \mathbb{Z}_q^*$, and computes a point $R_{SP} \in G$ such that $R_{SP} = zP$. Next, the SP also generates $M_{SP} = R_{SP} + t_{SP} PK_{SP}$, where $t_{SP} = h(T_{SP})$ and T_{SP} denotes the SP's current timestamp. Note that, the point R_{SP} serves as the Diffie-Hellman key parameter as well.

4. Therefore, the SP can negotiate the session key $SK_{SPi} = z\hat{R}_i = zw_iP$ for protecting the subsequent communications.

5. To avoid any modification of the message, the SP exploits the ECDSA signature, which is also adopted by current standard for VANETs [9], to assure the integrity. The values $\{M_{SP}, T_{SP}\}$ are defined as the signed content in ECDSA signature to produce the signature σ_{SP}.

6. Finally, based on the response packet format shown in 4.3.3, the SP sends the values $<M_{SP}, T_{SP}, \sigma_{SP}>$ by back to the requesting vehicle.

***Batch verification and key agreement*:** Given n distinct requests denoted as $<ID_1, M_1, F_1, T_1>$, $<ID_2, M_2, F_2, T_2>$,..., $<ID_n, M_n, F_n, T_n>$ sent from $V_1, V_2, ..., V_n$, respectively. Similar to the verification of a single request, the following steps are performed by the SP.

1. To withstand replay attacks, we assume that the receiving time is T_{SP_now}. The SP checks whether $\triangle T \geq T_{SP\_now} - T_i$ is valid, where $\triangle T$ is the predefined endurable transmission delay. If yes, then go on; otherwise, the SP ceases this connection.

2. To ensure the validity, the SP calculates $CRK_i = h(ID_i \| v)P$ depended on the pseudo identity ID_i, for $1 \leq i \leq n$, public parameters as well as his own private secret v, and computes $t_i = h(T_i)$. With the CRK_i and t_i, the SP individually extracts the random point $\hat{R}_i = M_i - t_i CRK_i$, where $\hat{R}_i = (\hat{x}_i, \hat{y}_i)$ for $1 \leq i \leq n$.

   Up to now, the steps are the same as that of single request verification. The following steps are designed for batch verification and key agreement.

3. To verify a batch of requests, the SP accumulates $\sum_{i=1}^{n} F_i = \sum_{i=1}^{n} (mrk \cdot x_i)P = h(v||SID)(\sum_{i=1}^{n} x_i)P$, and computes $(\sum_{i=1}^{n} \hat{x}_i)$ to verify whether $\sum_{i=1}^{n} F_i \stackrel{?}{=} h(v||SID)(\sum_{i=1}^{n} \hat{x}_i)P$ is valid or not. If so, then go to next step; otherwise, this connection is terminated.

4. For mutual authentication, the SP picks a random number $z \in \mathbb{Z}_q^*$, and produces a point $R_{SP} \in G$ such that $R_{SP} = zP$. Next, the SP also generates $M_{SP} = R_{SP} + t_{SP} PK_{SP}$, where $t_{SP} = h(T_{SP})$ and T_{SP} denotes the SP's current timestamp. Note that, the point R_{SP} is still implied as the Diffie-Hellman key parameter.

5. The SP negotiates the session keys $SK_{SPi} = z\hat{R}_i = zw_iP$ with V_i, where $1 \leq i \leq n$, for protecting the subsequent communications. Note that the session keys are distinct because of the different w_i sent from different vehicles.

Material Message (M)	Timestamp (T)	Signature (σ)
20 bytes	4 bytes	56 bytes

Figure 4.3.3: The response packet format

6. Thanks to the ECDSA signature, the SP can generate only a single message to broadcast for a batch of the requesting vehicles. Each vehicle can verify the signature by the SP's public key PK_{SP} to assure the validity of SP and integrity of message. The values $\{M_{SP},\ T_{SP}\}$ are still defined as the content signed by the SP's private key RK_{SP} to produce the signature σ_{SP}.

7. Finally, following the response packet format, the SP broadcastly sends $<M_{SP},\ T_{SP},\ \sigma_{SP}>$ back to vehicles.

To be precise, the response packet format, shown in Figure 4.3.3, is consisted of the material message, timestamp and ECDSA signature[2]

[2]In ABAKA, we adopt the ECDSA-224, which is also recommended by the current VANET standard [9].

4.3.3.3 *Mutual Authentication and Key Agreement Procedures*

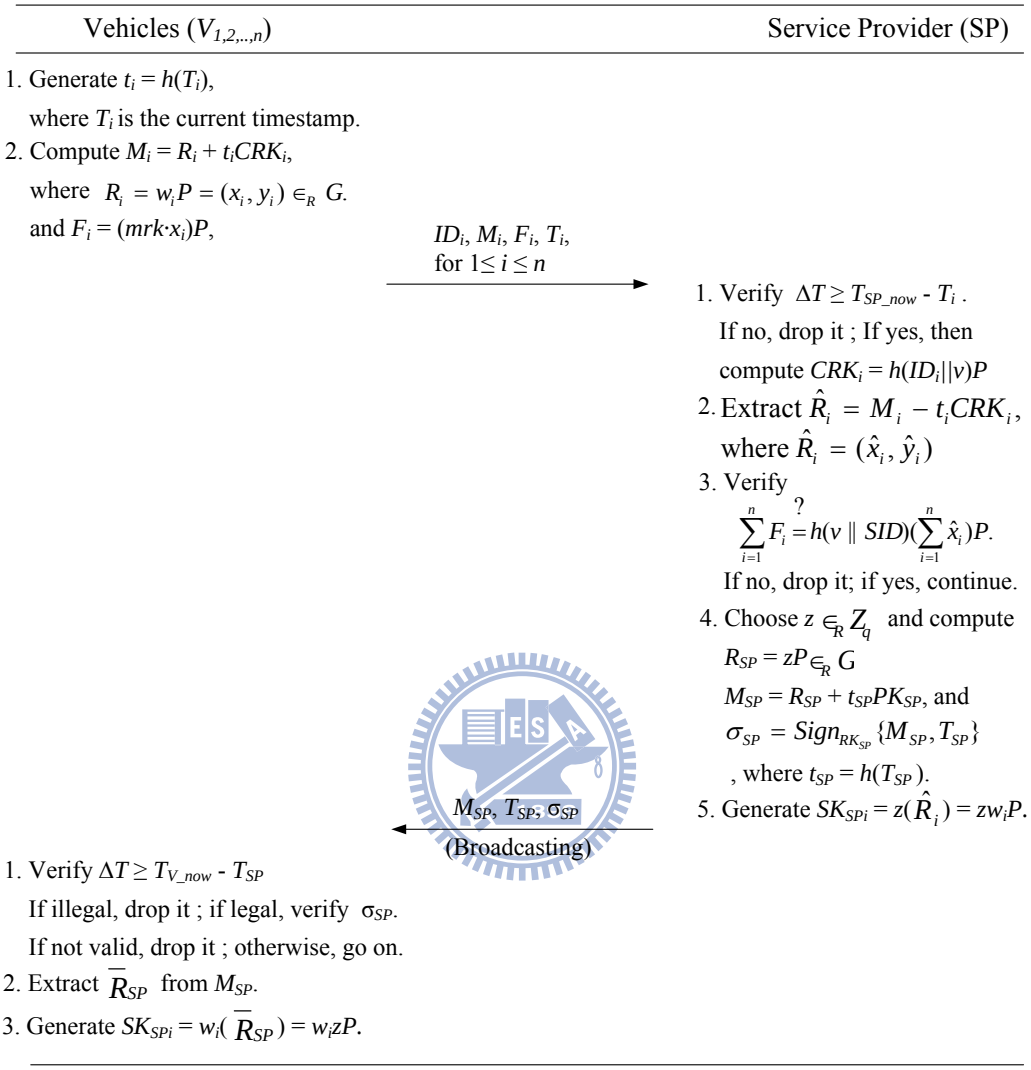


Figure 4.3.4: The ABAKA scheme

Given the response packet $< M_{SP},\ T_{SP}, \sigma_{SP} >$ sent from the SP, V_i carries out the following steps to mutually authenticate the validity of the SP, and to negotiate a session key for the confidentiality of the subsequent communications.

1. For freshness, V_i checks whether $\triangle T \ge T_{V\_now} - T_{SP}$ is valid, where $\triangle T$ is the predefined endurable transmission delay and T_{V_now} is the V_i's receiving time. If not, this session is dropped; otherwise, V_i first verifies the signature σ_{SP} to ensure

the integrity of the message. If σ_{SP} is legal, V_i goes to next step; otherwise, this connection is terminated.

2. For key agreement, V_i computes $\overline{R}_{SP} = M_{SP}\text{-}t_{SP}PK_{SP}$, where $t_{SP} = h(T_{SP})$, and generates the session key $SK_{SPi} = w_i\overline{R}_{SP} = w_izP$ to encrypt the messages in the subsequent communications.

Figure 4.3.4 summerizes the process of ABAKA scheme.

4.3.4 Discussion

4.3.4.1 Reliability Analysis

In the section, we discuss the reliability of ABAKA. Thanks to the batch verification, ABAKA enjoys several advantages such as lower verification delay and transmission overhead. However, the expense of the batch verification is that once an invalid request exists in a batch of requests, the batch verification may lose its efficacy. Note that the invalid request could come from a variety of reasons, such as packet loss, wireless channel interference, or the involvement of malicious attackers. This problem commonly accompanies with other batch-based verification schemes [5, 38]. To deal with this problem, we carefully analyze what happen if the problem occurs.

First, we develop a probabilistic model to characterize the risk that some requesting vehicles suffering from packet loss or sending bogus messages to pass the batch authentication based on the following assumptions:

- According to [8], the average packet loss ratio is almost lower than 0.07 % while the velocity of vehicles is changing from 10m/s-40m/s (36km/hr-144km/hr). On the other hand, if an attacker plans to send a bogus message at will, RSUs can rule out the bogus message if the signature verification of the message is failed. A possible case is that an attacker uses his valid private key issued by TA to sign a bogus message designed for passing the SP's batch authentication. In this case, RSUs will forward the bogus message to the aimed SP. However, once the attacker is detected by the SP, the SP can inform the TA to revoke the attacker's certificate, which can prevent

the attacker from sending a bogus message in the future. By above two protection mechanisms, we assume that at most 1% registered vehicles[3] can be compromised and send an invalid message passing the signature verification of RSUs to the SP in a batch period. While the number of registered vehicles (denoted as N_{Reg}) is assumed to be 10^4, the most number of compromised vehicles (denoted as N_C) is $N_{Reg} \times 1\% = 10^4 \times 1\% = 100$. For ease of analysis, we assume that a vehicle at most sends a request in a batch period.

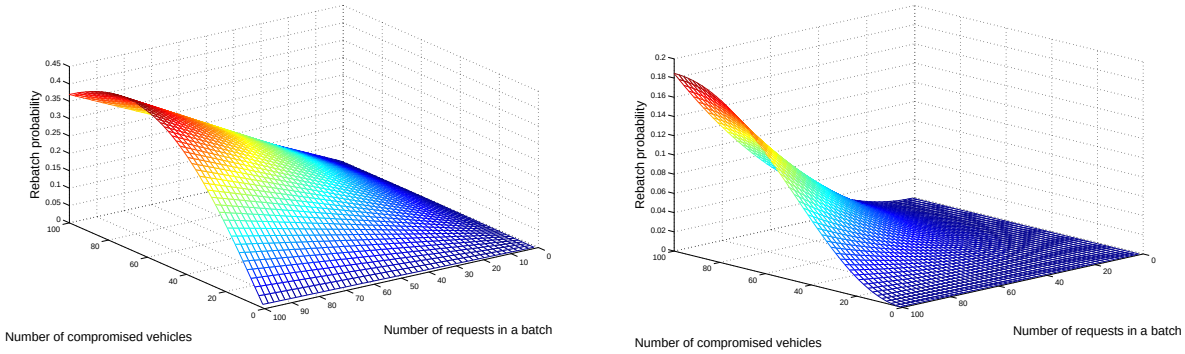
- In a period, the number of requests that an SP can process in a batch authentication is defined as N_B. Then, when one or more malicious requests within N_B, the other requests in the same batch are needed to re-authenticate which is referred to as rebatch authentication in this chapter.

Let $Pr\{i\}$ represent the probability that exactly i invalid requests sent from N_C are sending to the SP. The probability follows the hypergeometric distribution $\mathcal{H}(i, N_{Reg}, N_C, N_B)$ as follows.

$$Pr\{X = i\} = \frac{\binom{N_{Reg} - N_C}{N_B - i}\binom{N_C}{i}}{\binom{N_{Reg}}{N_B}}, \quad i = 0, 1, ..., 100$$

That is, in a period there are N_B requests to be authenticated, i invalid requests sent from N_C , and $N_B - i$ valid requests sent from $N_{Reg} - N_C$. Let A be the event that rebatch verification is required to successfully verify all valid requests N_B. Then, $Pr\{A\}$ can be represented as

[3] In [7], the attacker can compromise at most 0.2% entities subordinated by the TA.



(a) $i = 1$ (b) $i = 2$

Figure 4.3.5: Rebatch probability in ABAKA under different N_C and different N_B, where $1 \leq N_C \leq 100$, $1 \leq N_B \leq 100$

$$
\begin{aligned}
& Pr\{A\} \\
& = Pr\{i = 1\} + ... + Pr\{i = 100\} \\
& = \frac{\begin{pmatrix} N_{Reg} - N_C \\ N_B - 1 \end{pmatrix} \begin{pmatrix} N_C \\ 1 \end{pmatrix} + ... + \begin{pmatrix} N_{Reg} - N_C \\ N_B - 100 \end{pmatrix} \begin{pmatrix} N_C \\ 100 \end{pmatrix}}{\begin{pmatrix} N_{Reg} \\ N_B \end{pmatrix} \qquad\qquad \begin{pmatrix} N_{Reg} \\ N_B \end{pmatrix}} \\
& = \frac{\sum_{i=1}^{100} \begin{pmatrix} N_{Reg} - N_C \\ N_B - i \end{pmatrix} \begin{pmatrix} N_C \\ i \end{pmatrix}}{\begin{pmatrix} N_{Reg} \\ N_B \end{pmatrix}}
\end{aligned}
$$

That is, there is at least an invalid request in a batch, which leads to the failure of a batch verification. Hence, rebatch verifications are required. Then, we demonstrate the relationship between the number of compromised vehicles and that of requests in a batch in Figure 4.3.5. In Figure 4.3.5, the number of compromised vehicles is assumed to be 0-100, and a batch verification can simultaneously authenticate 0-100 requests. We can observe that the probability of rebatch verification is at most about 0.42 while only one invalid request ($i = 1$) in a batch, and dramatically drops to 0.18 while there are two invalid requests ($i = 2$) in the batch. The probability is almost negligible, approximately lower than 0.06 while there are more than two invalid requests ($i \geq 3$) in a batch. To tackle

Algorithm 4.1 Detection algorithm

- Data: The SP received a batch of requests BR=$\{Req_1, Req_2, ..., Req_n\}$
- Result: Output the invalid requests if there are invalid requests in RB; otherwise, return $Ture$.

```
1 DetAlg(BR):
2   begin
3     if BatchVerify(BR) then
4         return True;
5     else if Num(BR) == 1 then
6         return ID_i∈BR as an invalid request;
7     else
8         set BR_Front = {Req_1, Req_2, ..., Req_⌈n/2⌉};
9         set BR_Rear = {Req_⌈n/2⌉+1, Req_⌈n/2⌉+2, ..., Req_n};
10        DetAlg(BR_Front);
11        DetAlg(BR_Rear);
12    end if
13  end
```

the invalid request problem, we further propose a detection algorithm to find the invalid request in next subsection. Based on the proposed detection algorithm, we discuss the cost of a rebatch verification in Section 4.5.3. Moreover, we examine the expected verification delay including the original verification cost and the expected rebatch verifications cost in Section 4.5.4. The result shows that ABAKA can enjoy the more efficient than conventional ECDSA and other batch-based schemes even if the invalid request problem exists.

4.3.4.2 Invalid Request Detection

In Section 4.3.4.1, we discuss the probability of rebatch authentication. In this section, we provide a detection algorithm for detecting invalid requests. The concept of detection algorithm is based on "divide-and-conquer" approach [39]. When failing to verify a batch of requests, the SP can divide the batch into several subbatchs, and then separately check the validity of each subbatch. If the number of requests in a subbatch lefts only one and the request still remains invalid, then the SP determines that this request in the subbatch is invalid. The detection algorithm with binary divisions is shown in Algorithm 1.

4.4 Security analysis

As mentioned in Section 4.2.3, we analyze the security objectives of the proposed ABAKA as follows.

- Mutual authentication: ABAKA achieves mutual authentication between the service provider (SP) and requesting vehicles based on Elliptic Curve Discrete Logarithm Problem (ECDLP) and ECDSA certificates. To be authenticated by the SP, a requesting vehicle V_i must be able to produce the corresponding private key CRK_i to conceal the random point R_i into M_i and to generate the valid verification message $F_i = (mrk \cdot x_i)P$, where x_i is the x coordinate of point R_i. Without knowing the corresponding private key CRK_i, it is computationally infeasible to forge a valid pair (M_i, F_i). By only knowing M_i and t_i, it is still difficult to obtain R_i and CRK_i. That is, $M_i = R_i + t_i CRK_i$ is a Diophantine equation, which is also the security fundamental of IBV scheme [5]. Without knowing R_i and mrk, attackers cannot counterfeit the verification message F_i as well, which is based on ECDLP. Notice that, even if insider attackers who are the legitimate users are trying to impersonate other users, the insider attack is also withstood by the difficulty of ECDLP. Since the insider attacker neither realize the random point R_i nor acquire the corresponding private key CRK_i, the insider attackers are not able to forge a valid F_i. On the other side, the authenticity of the SP is guaranteed by ECDSA, which is also adopted as the current standard [9] in VANETs.

- Session key establishment: For confidentiality of subsequent communications between the SP and requesting vehicles, ABAKA provides the capability of negotiating session keys with vehicles. We exploit the concept of Elliptic Curve Diffie-Hellman (ECDH) key exchange protocol [34] to establish the session keys. The random points $R_i = w_i P$ and $R_{SP} = zP$ serve as the exchange key parameters chosen by the requesting vehicles and SP, respectively. Only the legitimate SP and vehicles are able to compute the session key $SK_{SPi} = z(R_i) = zw_i P = w_i zP = w_i(R_{SP})$, which is also relied on the difficulty of ECDLP. Moreover, the session key enjoys the perfect forward secrecy, where even if a long-term secret is compromised, the previous session keys

still remain confidential. In summary, ABAKA can securely negotiate a session key shared between the SP and each vehicle.

- Privacy Preservation: The privacy of each vehicle can be well-protected by the pseudo identities $ID_i = (ID_i^1 || ID_i^2)$, where $ID_i^1 = h(R_i)$, and $ID_i^2 = RVID \oplus H(v \oplus ID_i^1)$. ID_i^1 and ID_i^2 are made up of one-way hash function and XOR operation without leaking any identity information. Moreover, ABAKA achieves the conditional privacy, meaning that the SP should be able to realize who is accessing the services by computing $ID_i^2 \oplus H(v \oplus ID_i^1) = RVID \oplus H(v \oplus ID_i^1) \oplus H(v \oplus ID_i^1) = RVID$. Therefore, the requirement of conditional privacy preservation is met.

- Low transmission overhead and fast verification: In terms of transmission overhead, a requesting vehicle does not need the signature and corresponding public key certificate in each request message. Although the SP computes a signature and broadcastly sends to all requesting vehicles in the response packet, we can preload the service provider's public key into each vehicle to mitigate the transmission overhead. Note that the service provider's public key is fixed so we can preload it in the system initiation phase [12]. Moreover, we can only broadcast the signature to a few RSUs, in where there are requesting vehicles, instead of sending to all vehicles. As for the fast verification, ABAKA adopts the concept of the batch verification to simultaneously authenticate a batch of requests. The more requests come, the more performance advantages of our scheme emerge, which is demonstrated in Section 4.5.1.

Proposition 1. *A batch verification is successful if and only if all individual requesters are valid.*

($\Longrightarrow$) If a batch verification is successful then all individual requests are valid. Because a batch verification is successful, $\sum_{i=1}^{n} F_i = \sum_{i=1}^{n} (mrk \cdot x_i) P = h(v||SID)(\sum_{i=1}^{n} \hat{x}_i) P$ is held. By tamper-proof device, the value of $mrk = h(v||SID)$ can be derived by only the SP and P is the public generator of the cyclic additive group G meaning that it is not easy to forge. Then, it represents that $\sum_{i=1}^{n} \hat{x}_i$ is valid. Each $\hat{x}_i$ is the x coordinate of the point $\hat{R}_i$ derived from M_i sent by V_i. The bits of each $\hat{x}_i$ is at least 224 bits meaning that the probability

that an attacker can guess a correct $\hat{x}_i$ with corresponding M_i is extremely low more than $\frac{1}{2^{224}}$. Note that because each $\hat{x}_i$ is the x coordinate of the point $\hat{R}_i$, the value of $\hat{x}_i$ cannot be tampered at will. As a result, it is reasonable to infer that each individual request is valid if a batch verification is successful.

($\Longleftarrow$) If all individual requests are valid then a batch verification is successful. As long as each individual request is valid, each tuple (ID_i,M_i,F_i,t_i) can be correctly verified by the formula $F_i \stackrel{?}{=} (h(v||SID)\cdot\hat{x}_i)P$, where $\hat{x}_i$ is the x coordinate of the point $\hat{R}_i$ and $\hat{R}_i = M_i - t_i\cdot h(ID_i||v)P$. Then, we can accumulate all individual F_i into $\sum_{i=1}^{n} F_i = h(v||SID)(\sum_{i=1}^{n} x_i)P$. As a result, the formula $\sum_{i=1}^{n} F_i = h(v||SID)(\sum_{i=1}^{n} x_i)P \stackrel{?}{=} h(v||SID)(\sum_{i=1}^{n} \hat{x}_i)P$ will hold since each x_i is the same as $\hat{x}_i$. Note that $\hat{x}_i$ is extracted by the SP from the receiving M_i.

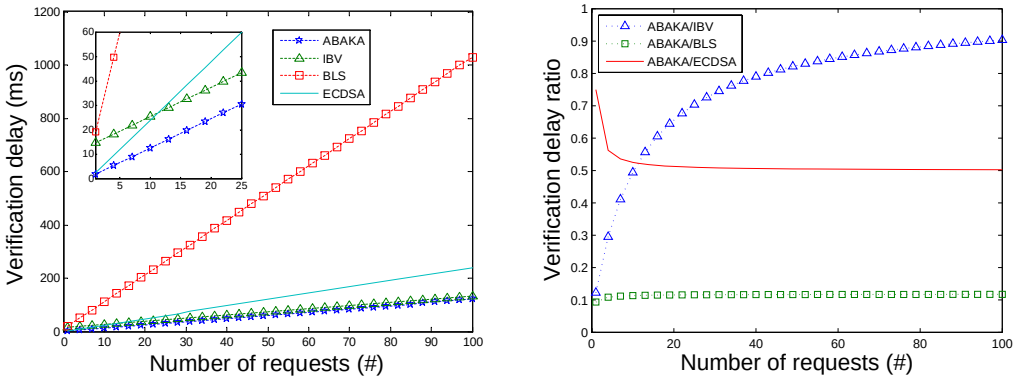
4.5 Performance Evaluations

In this section, we first evaluate the performance of ABAKA in terms of the verification delay, transmission overhead, and verification cost for rebatch verifications by analytical analysis. In [6], an authentication and key establishment protocol with ECDSA signature scheme, referred to as ECDSA-AKA scheme in this chapter, has been proposed. Here, we compare ABAKA with some related protocols, such as IBV [5], BLS [40, 41], and ECDSA-AKA [6]. Note that ECDSA is the current standard signature algorithm adopted by IEEE 1609.2 [9][4], while IBV and IBS are notable batch-based verification schemes. Next, we further verify the efficiency and applicability of the proposed ABAKA in real-world environment using ns-2 [29]. In addition, to full estimate the road environment and vehicular traffic, a well-known mobility model generation tool, TraNS [30], is adopted in the simulation.

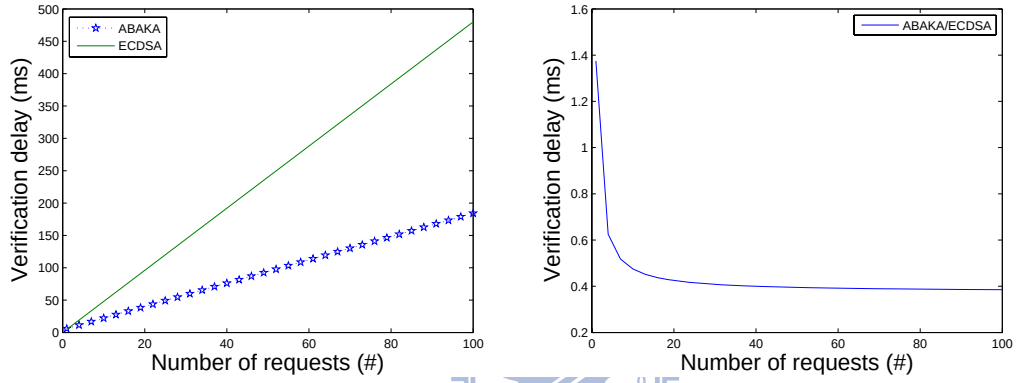
4.5.1 Verification Delay

Firstly, we define the time complexity of the cryptographic operations required in ABAKA and other schemes. Let T_{mul} denote the time to perform one point multiplication over

[4]In DSRC standard, the ECDSA is used for the message verification. Considering to mutual authentication and key agreement, the ECDSA-AKA [6] can be employed.



(a) One-way authentication case



(b) Mutual authentication case

Figure 4.5.1: Verification delay vs. number of requests

an elliptic curve, T_{par} be the time to execute a pairing operation, and T_{mtp} represent the time of a MaptoPoint hash operation. Since the three operations dominate the speed of verification, we only consider the three operations and neglect the other operations such as additive and one-way hash function. Here, we adopt the experiment in [42] for an MNT curve [37] of embedding degree k=6 and 160-bit q. The implementation was executed on an Intel Pentium IV 3.0 GHz Machine. The following result are obtained: T_{mul} is 0.6 ms, and T_{par} is 4.5 ms. T_{mtp} takes the same time as T_{mul}.

Table 4.2 shows the verification delay of all schemes in terms of authenticating a single request and n requests. Notice that IBV, BLS are designed for the message verification without mutual authentication and key agreement. For fairness, we compare the verification delay of ABAKA, IBV, BLS as well as ECDSA-AKA schemes in one-way authentication

Table 4.2: Comparisons of verification delay (ms)

	Authenticate a single request		Authenticate n requests	
	Vehicle auth.	SP auth./key agree.	Vehicles auth.	SP auth./key agree.
ABAKA	$3T_{mul}$	$8T_{mul}$	$(2n+1)T_{mul}$	$(n+7)T_{mul}$
IBV	$3T_{par} + T_{mtp} + T_{mul}$	N./A.	$3T_{par} + nT_{mtp} + nT_{mul}$	N./A.
BLS	$4T_{par} + 2T_{mtp}$	N./A.	$(2n+2)T_{par} + 2nT_{mtp}$	N./A.
ECDSA-AKA	$4T_{mul}$	$4T_{mul}$	$4nT_{mul}$	$4nT_{mul}$

case.[5] Then, we discuss the verification delay of ABAKA and ECDSA-AKA schemes since both two schemes provide the functionality of mutual authentication and key agreement. Figure 4.5.1 (a) shows the effect on verification delay of all schemes in one-way authentication case while the number of requests increases. For clarity, we zoom in the number of request ranging from 0 to 25 in the embedded small figure. Furthermore, we also show the ratio of the verification delay for comparison. Figure 4.5.1 (b) focuses on mutual authentication case to compare the verification delay of ABAKA and ECDSA-AKA. From Figure 4.5.1 (a), we can observe that ABAKA holds significant advantages compared to the other schemes. The ratio of verification delay shows that ABAKA is almost constantly 89 % faster than BLS. ABAKA is 48% faster than ECDSA when the number of requests is larger than 10. It is worth to mention that IBV can verify a batch of numerous messages as almost fast as ABAKA since the verification delays of both two schemes have fewer relationships with the number of requests.

4.5.2 Transmission Overhead

In this section, we analyze the transmission overhead of ABAKA compared to that of IBV, BLS and ECDSA-AKA schemes. The transmission overhead is consisted of two aspects: the transmission overhead incurred by delivering the packet from requesting vehicles to the service provider (V to SP), and from the service provider to the requesting vehicles (SP to V). Table 4.3 lists the total transmission overhead of all schemes in terms of sending a single request and n requests. The packet size of ABAKA, defined in Figure 4.3.2 and 4.3.3, cost 84 and 80 bytes, respectively. Note that ABAKA utilizes the advantage of broadcasting

[5]The message verification can be regarded as one-way authentication since the signature can be used to manifest the identity of user. However, one-way authentication does not provide the functionality of session key agreement.

Table 4.3: Comparisons of transmission overhead (bytes)

	Send a single request		Send n requests	
	V to SP	SP to V	V to SP	SP to V
ABAKA	84 bytes	80 bytes	$84n$ bytes	$N_{RSU}\times 80$ bytes
IBV	63 bytes	N./A.	$63n$ bytes	N./A.
BLS	146 bytes	N./A.	$146n$ bytes	N./A.
ECDSA-AKA	167 bytes	167 bytes	$167n$ bytes	$167n$ bytes

N_{RSU} is the number of roadside units (RSUs) where the requests sent from.

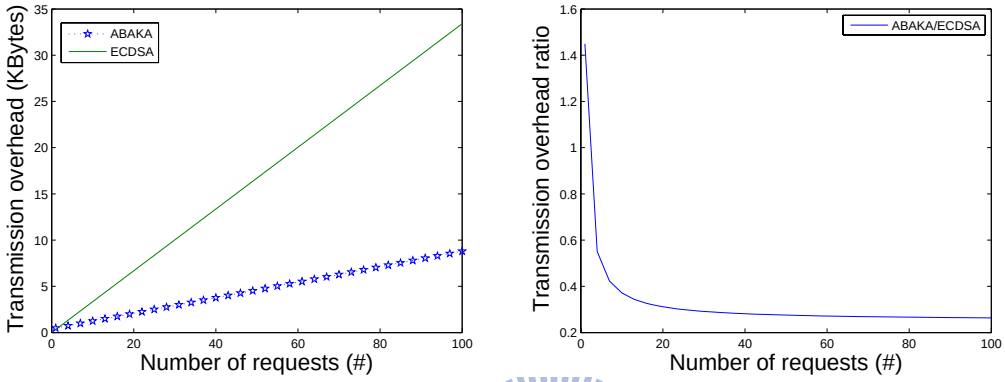


Figure 4.5.2: Transmission overhead vs. number of requests (in mutual authentication case)

to design the response message.[6] Therefore, the transmission overhead from SP to V in ABAKA can be much lower than traditional scheme. The packet of IBV is consisted of a 21-byte signature and a 42-byte pseudo identity. The packet of BLS and ECDSA are comprised of a signature and a 125-byte certificate, but BLS adopted a short signature cutting down the signature size from 42 to 21 bytes. In Figure 4.5.2, we also discuss the transmission overhead of ABAKA and ECDSA-AKA schemes [6] while a number of requests pour in. From Figure 4.5.2, we can see that the ratio of the transmission overhead sharply drops from 138% to 31% when the number of requests is over 10. More precisely, as long as the number of requests is more than 13, the transmission overhead of ABAKA is lower 68.9% than that of ECDSA-AKA.

[6] According to [1] and [5], there could be roughly 180 vehicles in the communication range of an RSU in a high density traffic scenario. It is highly possible that several requests from the same RSU can be verified in the same batch. So, we assume $N_{RSU} = 5$, where N_{RSU} is the number of RSUs where the requests sent from, in the following analysis.

Table 4.4: Verification cost for rebatch verification

	First batch verification	Rebatch verification
ABAKA	$(2n+1)T_{mul}$	$1T_{mul}$
IBV	$3T_{par} + nT_{mtp} + nT_{mul}$	$3T_{par}$
BLS	$(2n+2)T_{par} + 2nT_{mtp}$	$(n+1)T_{par}$

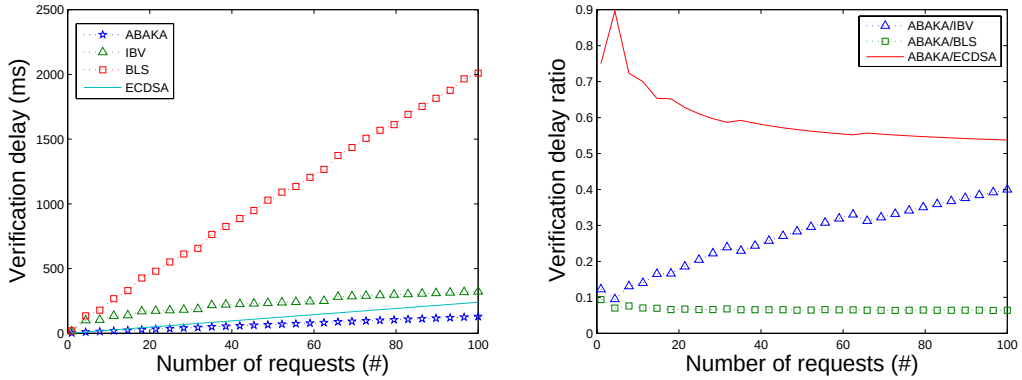
4.5.3 Verification Cost for Rebatch Verifications

According to the reliability analysis described in Section 4.3.4.1, the probability of only one invalid request in a batch is the most significant. Here, we further analyze the cost for a vehicle if rebatch verifications are required. We elaborately analyze the verification delay for a vehicle in the three batch-based schemes, ABAKA, IBV, and BLS, in Table 4.4. In ABAKA, the SP has to calculate requesting vehicles' corresponding private key CRK_i and t_iCRK_i to derive random point R_i in the first batch verification, which takes 2 T_{mul}. However, in rebatch verification, the SP only spends 1 T_{mul} for verification.[7] In IBV and BLS, some operations can be omitted in the rebatch verification as well. In the following analysis, we assume that the number of requests in a batch is 100 ones. To be precise, we provide the verification cost in the worst case and average case. Although the ECDSA scheme is not required to perform the rebatch verification, we also show the verification cost of ECDSA in the following cases to examine the value of the batch-based schemes.

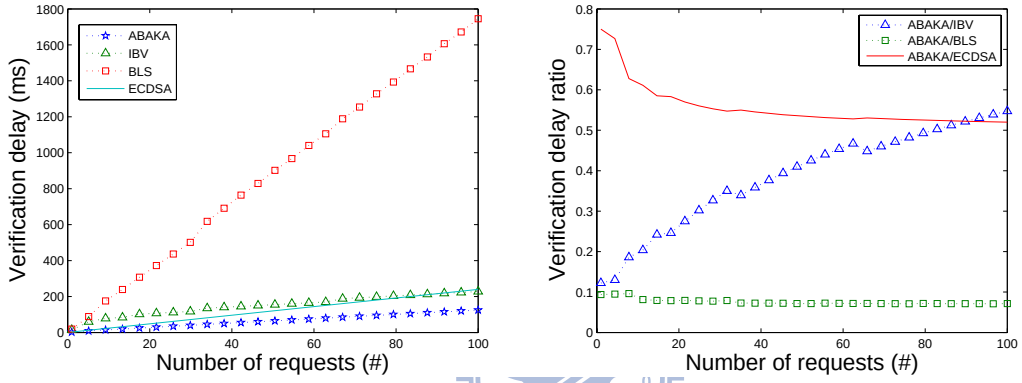
- Worst Case: According to the proposed detection algorithm in Section 4.3.4.2, the worst case means that a valid request is always with the invalid request in the same batch until the last batch division. A batch of requests can be divided at most $\lceil log_2 n \rceil$ times, where n is the number of requests in a batch. Let $T_{first\_ver.}$ denote the time to perform the verification in the first time, and $T_{rebatch\_ver.}$ denote the time to perform the verification in a rebatch verification. As a result, the total verification delay for a valid request in worst case is

$$T_{worst} = 1 \times T_{first\_ver.} + 2 \times \lceil log_2 n \rceil \times T_{rebatch\_ver.}$$

[7] Similar to [5], we only concern the cost of three dominant operations, T_{mul}, T_{par}, and T_{mtp}.



(a) Verification delay for rebatch verifications in worst case



(b) Verification delay for rebatch verifications in average case

Figure 4.5.3: Verification delay for rebatch verifications vs. number of requests

- Average Case: The average case is the total verification delay over all possible cases divided by the number of possible cases. Then, the total verification delay for a valid request in the average case is

$$T_{Avg} = 1 \times T_{first\_ver.} + \frac{1}{\lceil log_2 n \rceil + 1} \sum_{i=1}^{\lceil log_2 n \rceil} (T_{first\_ver.} + 2 \times T_{rebatch\_ver.})$$

Figure 4.5.3 (a) shows the verification delay for rebatch verifications in worst case while the number of requests in a batch is changed from 1 to 100 requests. Along with the verification delay for rebatch verifications, the comparison of the ratio of verification delay for rebatch verifications is also represented. And, the average case is demonstrated in Figure 4.5.3 (b). From Figure 4.5.3, we can observe that ABAKA outperforms the other schemes, even ECDSA. Note that ECDSA is not batch-based verification without additional rebatch

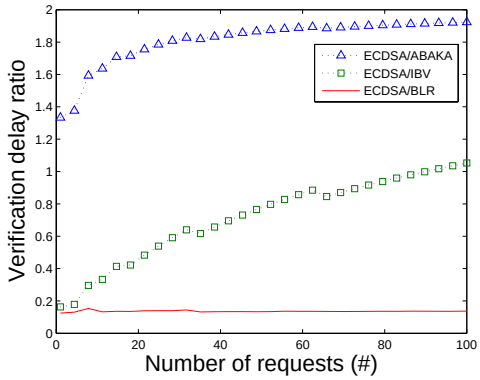


Figure 4.5.4: Verification delay ratio compared with ECDSA scheme vs. the number of requests

verification delay. In Figure 4.5.3 (a), ABAKA is almost constantly faster than BLS by 94%, outperforms IBV at least 60 %, and gains about 41% faster than ECDSA while the number of requests is more than 28. As compared with BLS and ECDSA, ABAKA enjoys the more advantages while the more requests are issued. In the average case, the advantage of ABAKA over ECDSA is more significant. It is worth to mention that IBV also enjoys the advantages, faster than ECDSA, in the average case while the number of requests is more than 94. The reason is that both ABAKA and IBV take the constant time to verify a batch of requests but ECDSA and BLS do not. Moreover, in order to measure the effectiveness of the batch-based schemes, we compare the ratio of the verification delay of the three batch-based schemes with ECDSA scheme in Figure 4.5.4. As compared with ECDSA, only ABAKA can have the better performance than that of ECDSA no matter how many numbers of requests appear. When the number of requests is up to 90, IBV outperforms than ECDSA. Unfortunately, BLR did not have the performance advantage. As a result, ABAKA should be more suitable than other batch-based schemes in VANETs.

4.5.4 Expected Verification Delay

Based on the reliability analysis described in Section 4.3.4.1 and verification cost for rebatch verifications described in Section 4.5.3, we further examine the expected verification delay composed of the original verification cost and the expected verification cost for rebatch verifications. Here, we consider the worst case of rebatch verifications. Let $T_{orignal\_ver.}$

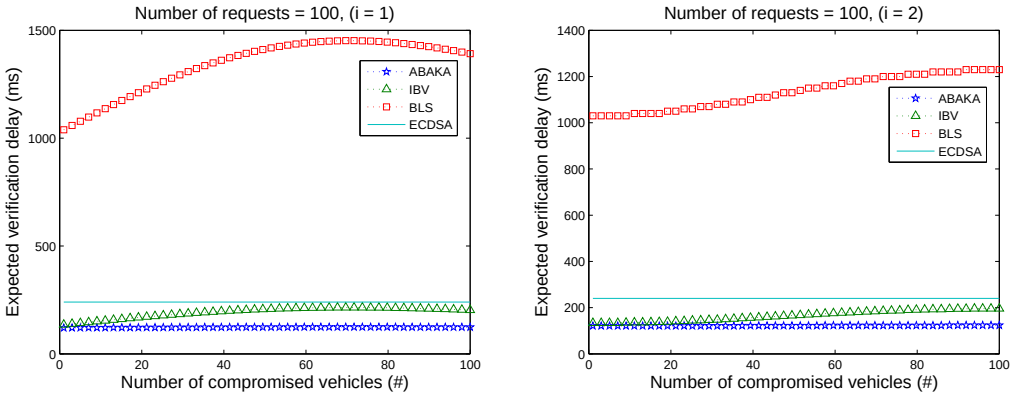
denote the time to perform the unidirectional verification in the proposed scheme without any invalid request, $Pr_{rebatch}$ denote the probability of performing the rebatch verification, and $T_{rebatch\_cost}$ represent the extra verification cost for rebatch verifications. Thus, the expected verification delay, denoted as $T_{expected}$, can be formulated as below.

$$T_{expected} = T_{orignal\_ver.} + Pr_{rebatch} \times T_{rebatch\_cost}$$

In Figure 4.5.5, the relationship between the expected verification delay and the different number of compromised vehicles is presented while the number of requests in a batch is set to 100. With the results of the reliability analysis, we examine the most possible two cases, there are one (i = 1) or two (i = 2) invalid requests sent from compromised vehicles in a batch of requests. We can observe that the variation of the number of compromised vehicles only slightly affects on the expected verification delay for ABAKA. The reason is that the rebatch verification cost for ABAKA is relatively less than other batch-based verification schemes. As compared with ECDSA without the rebatch verification cost, ABAKA also keeps the superior expected verification delay. Note that the rebatch verification cost for the (i = 2) case can be derived from Section 4.5.3 by assuming that the two invalid requests are separately distributed in the front part and the rear part of requests, which is the worst case. To sum up, it is anticipated that ABAKA could effectively ease the verification burden of SPs.

4.5.5 Simulation Evaluation

In this section, we adopt the ns-2 simulator [29] to properly estimate the real-world road environment and vehicular traffic. In order to genuinely generate the mobility of the real-word vehicles, we use the mobility model generation tool, named TraNS, introduced by [30]. TraNS can take advantage of the publicly available Topologically Integrated Geographic Encoding and Referencing (TIGER) database from the U.S. Census Bureau, where the street maps of cities/towns in the United States are offered. Our simulation adopts the



(a) $i = 1$ (b) $i = 2$

Figure 4.5.5: Expected verification delay vs. number of compromised vehicles

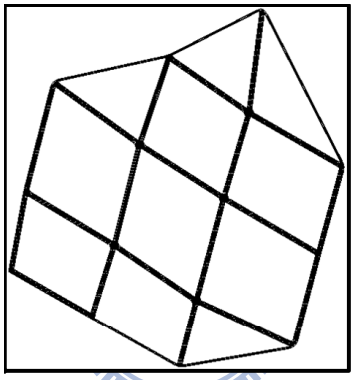


Figure 4.5.6: A city-street map

Table 4.5: Simulation Configuration

City simulation area	$1000m \times 1000m$
Communication range	250 m
Simulation time	100 s
Wireless Protocol	802.11a
Channel bandwidth	6 Mbs
Pause time	0 s
Packet size for ECDSA message	167 bytes
Packet size for ABAKA message	84 bytes

map showed in Fig 4.5.6, which corresponds to a part of Manhattan, New York, USA. At first, vehicles are randomly scattered on the roads and then move towards randomly selected intersections in the map. Vehicles are driving along the roads with a random speed between $1m/s$ to $40m/s$. The road-speed limit is also implemented in every street.

All possible cryptographic operations in the simulation are considered to have the same simulation delay. We assume that some 20% vehicles are requesting services, a value which is used to calculate the verification delay. In this simulation, we are interested in the performance of ABAKA and ECDSA-AKA since only the two schemes can provide mutual authentication and key agreement. All the simulation parameters are listed in Table 4.5.

The average message delay (denoted as $avgD$) and average loss ratio (denoted as $avgLR$) are considered in this simulation, and can be expressed as follows.

$$avgD = \frac{1}{N_A \cdot M^n_{sent} \cdot SP^n} \sum_{n \in A} \sum_{m=1}^{M^n_{sent} SP^n} \sum_{s=1} (T^{n\_m\_s}_{trans} + T^{n\_m\_s}_{v-auth} \cdot L^s + T^{n\_m\_s}_{SP-auth})$$

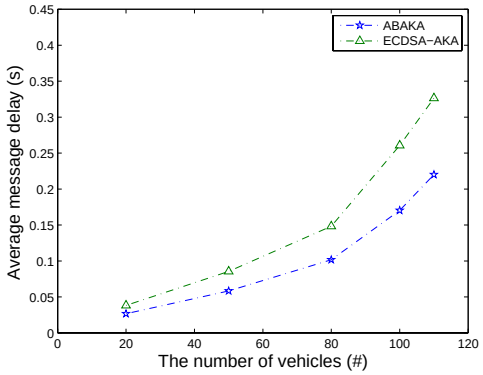
where A is the sample area in this simulation, N_A is the number of vehicles in A, M^n_{sent} is the number of request messages sent by vehicle n, and SP^n is the number of service providers where vehicle n has registered. For simplicity, we assume that $SP^n = 1$ in this simulation. $T^{n\_m\_s}_{trans}$ is the time that vehicle n transmits messages m to service provider s, $T^{n\_m\_s}_{v-auth}$ is the time that service provider s authenticates vehicle n which is triggered by message m, and $T^{n\_m\_s}_{SP-auth}$ is the time that vehicle n authenticates service provider s which is triggered by message m. n_m_s represents the message m sent by vehicle n and received by service provider s, and L^s is the length of the queue in service provider s.

$$avgLR = \frac{1}{N_A} \cdot \sum_{n=1}^{N_A} \frac{M^n_{consumed}}{\sum_{s=1}^{SP^n} M^n_{arrived}}$$

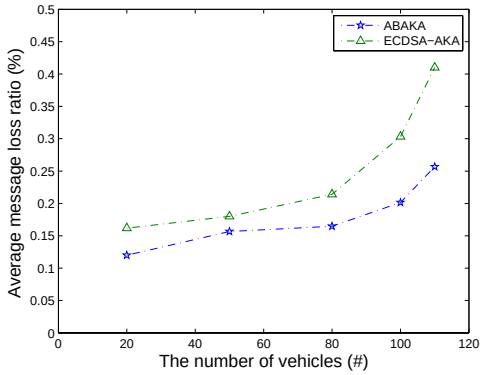
where $M^n_{consumed}$ means the number of messages consumed by vehicle n in the application layer, and $M^n_{arrived}$ represents the number of messages that received by the service provider s in the application layer.

4.5.5.1 Impact of Vehicle Density

In the first set of simulation, we investigate the impact of vehicle density. Figure 4.5.7 shows the simulation results on the average message delay and the average message loss rate. In general, the more vehicles appear, the more advantages ABAKA holds. In Figure 4.5.7 (a), ABAKA outperforms ECDSA-AKA between 31% and 34%. As you can see that the curve tendency of message delay corresponds to the analytical results analyzed in Section 4.5.1 and Figure 4.5.1 (b). Note that the analytic results do not include the transmission delay. With regard to the message loss ratio, both ABAKA and ECDSA-AKA increase the

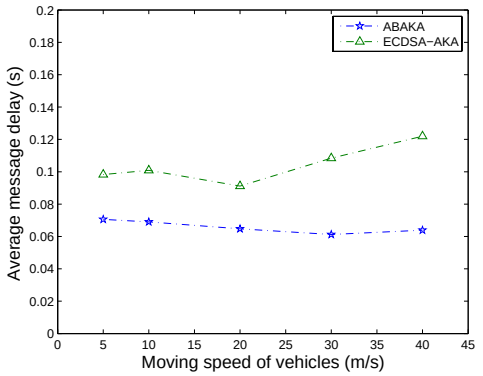


(a) Average delay vs. the number of vehicles

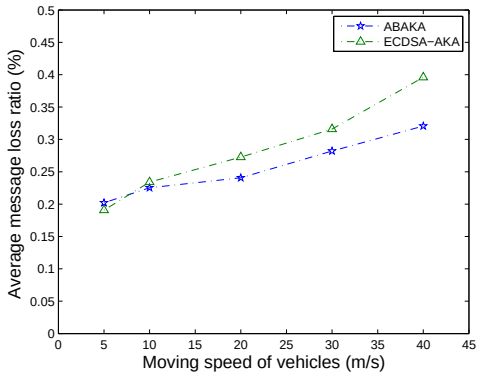


(b) Average loss ratio vs. the number of vehicles

Figure 4.5.7: Impact of vehicle density



(a) Average delay vs. moving speed of vehicles



(b) Average loss rate vs. moving speed of vehicles

Figure 4.5.8: Impact of vehicles' moving speed

message loss ratio while the number of vehicles soars. The increasing ratio of ABAKA is between 5% and 24%, and that of ECDSA-AKA is between 11% and 30%. As compared to ECDSA-AKA, ABAKA has the better performance in terms of message loss ratio, which has reached to 38% while the vehicle density is greater than 100.

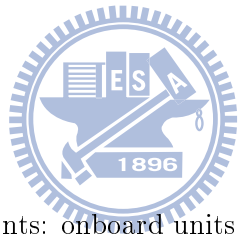
4.5.5.2 Impact of Vehicle Moving Speed

In the second set of simulations, the average speed of the vehicles is changed from $5m/s$ to $40m/s$ ($36km/hr$ to $144km/hr$). In this simulation, we assume that the number of vehicle is 50. The simulation results on the average message delay and average message loss rate are shown in Figure 4.5.8. As you can see that the average message delay of ABAKA is

slightly affected by the moving speed of vehicles. However, the average message delay of ECDSA-AKA is increased significantly as the moving speed is over $20m/s$. It can be seen that ABAKA yields less message delay than ECDSA-AKA does in every speed. In terms of the message loss ratio, we can see that both two schemes are significantly affected by the moving speed, particularly for ECDSA-AKA. The different moving speed of vehicles will trigger different times of handoff procedures, which are the process of transferring an ongoing session from one RSU to another. The handoff procedures may incur the higher message loss ratio while a data transmission is ongoing. From 4.5.8 (b), it is obvious that ABAKA is with lower message loss ratio while the moving speed is up to $20m/s$. The reason is that because the packet size of ABAKA is shorter than that of ECDSA-AKA, the period of packet transmission of ABAKA is also shorter than that of ECDSA-AKA. Therefore, we infer that the packet transmission of ECDSA-AKA has higher probability to be interrupted while the moving speed of vehicles is fast.

Chapter 5

PAACP : A Portable Privacy-Preserving Authentication and Access Control Protocol



5.1 Motivation

In VANETs, there are two components: onboard units (OBUs) and roadside units (RSUs). OBUs represent the wireless communication devices equipped in vehicles, and RSUs are wireless access devices located at critical points or intersections on the road. There are two kinds of communications: roadside-to-vehicle communications (RVCs) and inter-vehicle communications (IVCs). The birth of VANETs comes from improving the road safety. Therefore, safety-related applications are developed over VANETs. In addition to safety-related applications, VANETs also provide non-safety applications [43, 44] to offer maps, advertisements, and entertainment information [6]. For example, Microsoft Corp.'s MSN TV and KVH industries Inc. [45, 46] have introduced an automotive vehicle Internet access system, called TracNet, bringing Internet services to in-car video screen.

In the recent years, several researches on VANETs have been investigated by academic or industries, such as IEEE P1609.2 working group [9, 47] consortium. Most studies were interested in the performance of medium access control (MAC) layer or the routing issues

inherent in VANETs. Recently, some works addressed the security issues. As a special case of mobile ad hoc networks (MANET), VANETs may suffer any malicious user's behaviors, such as bogus information and replay attacks on the disseminated messages. Among various security threats, privacy preservation in VANETs is one of the new challenges to protect users' private information including the driver's name, license plate, model, and traveling route. In 2005, [48] first proposed a solution to tackle both security and privacy issues for safety-related applications. However, their solution is not complete and sound [49]. In 2007, [4] proposed a secure and privacy-preserving protocol, called GSIS, for VANET communications. GSIS adopted a group signature scheme in IVCs and ID-based cryptography (IBC) in RVCs to protect communication messages. All the above protocols were developed especially for safety-related applications. Similar to safety applications, non-safety applications in VANETs have to take both security and privacy issues into consideration [4, 12, 50]. In addition, designing a practical non-safety application for VANETs should take into consideration the following characteristics in VANETs [10, 49, 51].

1. Stringent time constraint in communication: The speed of a vehicle could be more than 140 km/h. The communication delay in IVCs or RVCs should be short enough to meet stringent time requirement [6, 9, 10].

2. Large scale networks: In general, with an inter-vehicle distance of 70 m, there are some 70 vehicles within a radius of 1 km around a given car. During a traffic jam, with an inter-vehicle instance of 5 m, there can be more than 1000 vehicles within the same region. Therefore, VANETs will be large scale networks [10, 51].

Both characteristics introduce performance and scalability issues in VANETs. In 2008, Zhang *et al.* [5, 6] proposed two schemes to deal with the scalability problem in VANETs. Wang *et al.* [49] proposed an enhanced communication protocol based on the infrastructure of [12, 48] to support non-safety applications with confidentiality and non-repudiation property. However, Wang *et al.*'s scheme did not address the scalability issue. Li *et al.* [10] also proposed a secure and efficient communication scheme with privacy preservation, called SECSPP, for non-safety applications in VANETs. Moreover, SECSPP discussed the security issue among service providers, roadside units and vehicles. In SECSPP, a vehicle

needs to acquire a blind signature for privacy preservation before the vehicle accesses the desired services from its neighboring RSU. A service provider (SP) is responsible for signing and verifying the validity of signatures, and also involves in session key establishment between the RSUs and requesting vehicles. There are some drawbacks in SECSPP:

1. Deficient in meeting stringent time requirement: When a vehicle tries to access a non-safety service via an RSU, the RSU must pass the signature sent from the requesting vehicle to the proper SP for verification, whereas the SP may be located in a distant network. The speed of a vehicle may be extremely high. It is possible that the response sent from the SP has not arrived yet, but the requesting vehicle had passed the transmission range of the RSU.

2. Lack of scalability in SP: All requests of non-safety applications must be first verified by the proper SP, which will become the bottleneck of SECSPP. The scalability issue rises in a popular SP if a large number of requests pours out.

3. Short of differentiated service access control: In SECSPP, when a vehicle sends the *Access_ Service_ Request* to an SP via an RSU, the SP only responds the accept/reject permission. However, in modern commerce model, an SP may provide several services with different access privileges for different users' requirements, named differentiated service access control [52].

The lack of scalability and access control in SECSPP will limit the development of non-safety applications. In this chapter, we propose a *P*ortable privacy-preserving *A*uthentication and *A*ccess *C*ontrol *P*rotocol, named PAACP, with the support of differentiated service access control. In addition, considering stringent time requirement in transmission delay, PAACP eliminates the communications between the roadside units (RSUs) and service providers (SPs). In a conventional access control scheme, SPs are usually responsible for determining the validity of the access requests. To get rid of the communication with SPs, we propose a novel portable access control method to store a portable service right list (SRL) into each vehicle, instead of keeping the SRLs in the SPs. In order to assure the validity and privacy of an SRL, we also propose a novel attachable blind signature. Based on the

attachable blind signature, vehicles (OBUs) cannot tamper the *SRL*. Therefore, PAACP can prevent privilege elevation attacks [53]. As for privacy protection of users, the SP cannot trace the current location of the requesting vehicle, due to the attachable blind signature and the no need of any verification by SP. In addition, PAACP is more efficient than conventional access control schemes since RSUs can verify the correctness of an *SRL* without backend communications with SPs. As a result, PAACP is desirable for large scale VANETs. To the best of our knowledge, PAACP is the first study supporting sophisticated service access control without the scalability problem in VANETs. In summary, PAACP achieves the following properties: 1) mutual authentication between the requesting vehicle and RSU, 2) dynamic session key establishment for the subsequent communications, 3) privacy preservation of the vehicle's information, 4) data confidentiality and integrity, 5) differentiated service access control, and 6) better scalability.

5.2 Related Work

5.2.1 Li et al.'s Work

Recently, Li *et al.* [10] proposed a secure and efficient communication scheme, named SECSPP, with authenticated key establishment for non-safety applications in VANETs. SECSPP is the first security scheme addressing non-safety applications with explicit authentication procedures [10]. In this section, we briefly introduce the procedures of SECSPP. The notations throughout Li et al.'s protocol are summarized in Table 5.1.

SECSPP consists of two phases: access authorization phase and access service phase. There are three participants: the vehicular node V_i, the service provider S_i, and the roadside device R_j. In the access authorization phase, V_i gets an authorized credential AC_i^* from S_i. Then, in the access service phase, V_i presents the authorized credential AC_i^* to access the desired services via R_j without disclosing any sensitive information .

5.2.1.1 Access Authorization Phase:

- Step 1: $V_i \rightarrow S_i$: $< VID_i,\ SID_i,\ T_{V_i},\ C \oplus (VID_i \| SID_i \| AC_i' \| M_i \| T_{V_i}) >$

Table 5.1: Notations of SECSPP

VID_i	The identity of vehicular node i
RID_j	The identity of roadside device node j
S_i	The identity of service provider i
VK_i	The secret key of V_i, based on non-interactive ID-based public key cryptography
RK_j	The secret key of R_j, based on non-interactive ID-based public key cryptography
SPK_i	The secret key of S_i, based on non-interactive ID-based public key cryptography
(PK_{S_i}, SK_{S_i})	The public key and private key of service provider S_i
MAC	The message authentication code $MAC=H(K; m)$, where m denotes message under the protection key of K.
M_i	The receipt of the service access sent from S_i for a user i to register as a legal user
$H(\cdot)$	A collision-free and public one-way hash function
$\oplus$	Exclusive OR operation
T_x	A timestamp, which node x attaches
$a\|b$	Concatenation of message a and b
$E_{PK_{S_i}}\{\cdot\}$	The asymmetric encryption function with service provider's PK_{S_i}
$D_{SK_{S_i}}\{\cdot\}$	The asymmetric decryption function with service provider's SK_{S_i}

First, V_i selects a random number a_1 and computes the authorized credential AC_i, where $AC_i=H(M_i\|VID_i\|a_1)$. Next, V_i chooses a blind factor a_2 to blind AC_i, and makes $AC'_i=a_2^{PK_{S_i}}\cdot AC_i$. Finally, V_i sends $<VID_i, SID_i, T_{V_i}, C\oplus(VID_i\|SID_i\|AC'_i\|M_i\|T_{V_i})>$ to S_i, where $C=(SID_i^2)^{H(T_{V_i})\cdot VK_i} \pmod N$ and VK_i is V_i's secret key, which is based on non-interactive ID-based public key cryptography [10].

- Step 2: $S_i \rightarrow V_i$:$<C'\oplus(SID_i\|VID_i\|AC''_i\|T_{S_i})>$

After receiving $<VID_i, SID_i, T_{V_i}, C\oplus(VID_i\|SID_i\|AC'_i\|M_i\|T_{V_i})>$, S_i reveals $(VID_i\|SID_i\|AC'_i\|M_i\|T_{V_i})$ by computing $C\oplus(VID_i\|SID_i\|AC'_i\|M_i\|T_{V_i})\oplus C'$ and verifies the validity of M_i, where $C'=(VID_i^2)^{H(T_{V_i})\cdot SPK_{S_i}} \pmod N$,[1] and SPK_{S_i} is the secret key of S_i. If is M_i valid, then S_i records (VID_i, M_i, T_{V_i}) in its database and marks M_i as non-fresh. In addition, S_i signs AC'_i with its private key SK_{S_i} by computing $AC''_i=AC_i^{'SK_{S_i}}=a_2\cdot AC_i^{SK_{S_i}}$. Finally, S_i delivers $<C'\oplus(SID_i\|VID_i\|AC''_i\|T_{S_i})>$ to V_i.

[1] Based on non-interactive ID-based public key cryptography, $C=(SID_i^2)^{H(T_{V_i})\cdot VK_i}=(VID_i^2)^{H(T_{V_i})\cdot SPK_{S_i}} \pmod N=C'$.

Once getting C'$\oplus(SID_i \| VID_i \| AC_i'' \| T_{S_i})$, V_i extracts $(SID_i \| VID_i \| AC_i'' \| T_{S_i})$ by calculating C'$\oplus(SID_i \| VID_i \| AC_i'' \| T_{S_i}) \oplus C$. Then, AC_i'' is unblinded by computing $AC_i'' \cdot (a_2)^{-1}$ and then V_i obtains $AC_i^* = AC_i^{SK_{S_i}}$. Moreover, AC_i^* is confirmed by checking whether $AC_i = (AC_i^*)^{PK_{S_i}}$. If yes, V_i believes AC_i^* is the signature of AC_i; otherwise, V_i drops it and stops this session.

5.2.1.2 Access Service Phase:

- Step 1: $V_i \rightarrow R_j$: $<Access\_Service\_Request, E_{PK_{S_i}}\{Access\_Service\_Request, RID_j, AC_i, AC_i^*, T_{V_i}, a_3\}>$

When a legal V_i wants to access the pay-service from the roadside unit R_j, V_i computes $E_{PK_{S_i}}\{Access\_Service\_Request, RID_j, AC_i, AC_i^*, T_{V_i}, a_3\}$, where a_3 is a random number generated by V_i. Then, V_i sends it with an $Access\_Service\_Request$ request to R_j.

- Step 2: $R_j \rightarrow S_i$: $<RID_j, T_{R_j}, C \oplus (E_{PK_{S_i}}\{Access\_Service\_Request, RID_j, AC_i, AC_i^*, T_{V_i}, a_3\})>$

Once receiving the $Access\_Service\_Request$ request sent from V_i, R_j computes $C \oplus (E_{PK_{S_i}}\{Access\_Service\_Request, RID_j, AC_i, AC_i^*, T_{V_i}, a_3\})$, where $C = (SID_i^2)^{H(T_{R_j}) \cdot RK_j}$, and delivers $(RID_j, T_{R_j}, C \oplus (E_{PK_{S_i}}\{Access\_Service\_Request, RID_j, AC_i, AC_i^*, T_{V_i}, a_3\}))$ to its back-end service provider S_i.

- Step 3:$S_i \rightarrow R_j$: $<SID, C$'$\oplus(Access\_Permission, a_3, b_1, AC_i, T_{s_i})>$

After receiving the message from R_j, S_i first extracts $E_{PK_{S_i}}\{Access\_Service\_Request, RID_j, AC_i, AC_i^*, T_{V_i}, a_3\}$ by computing $C \oplus (E_{PK_{S_i}}\{Access\_Service\_Request, RID_j, AC_i, AC_i^*, T_{V_i}, a_3\}) \oplus C$', where $C' = (RID_j^2)^{H(T_{R_j}) \cdot SPK_{S_i}} \pmod N$, and computes $D_{SK_{S_i}}\{E_{PK_{S_i}}\{Access\_Service\_R$ $RID_j, AC_i, AC_i^*, T_{V_i}, a_3\}\}$. Next, S_i will confirm the validity of the authorized credential AC_i^* by checking whether $AC_i = (AC_i^*)^{PK_{S_i}}$ holds or not. If yes, V_i is granted the access privilege from R_j and then S_i generates a random number b_1 and computes the temporary service key $TSK_i = H(a_3 \| b_1 \| AC_i \| 0)$; otherwise, this access request is denied. Last, S_i sends $<SID_i, C$'$\oplus(Access\_Permission, a_3, b_1, AC_i, T_{s_i})>$ to R_j.

- Step 4: $R_j \rightarrow V_i$:$<b_1, TSK_i \oplus (RID_j, b_2, T_{R_j})>$

Upon receiving the message from S_i, R_j acquires ($Access\_Permission$, a_3, b_1, AC_i, T_{s_i}) by computing C'$\oplus$($Access\_Permission$, a_3, b_1, AC_i, T_{s_i})$\oplus C$. Based on a_3, b_1, AC_i, R_j can compute TSK_i=$H(a_3\|b_1\|AC_i\|0)$ for the subsequent data encryption for accessing pay-services between V_i and R_j. Next, R_j generates a random number b_2 for mutual authentication and sends the message $<b_1$, $TSK_i\oplus(RID_j$, b_2, $T_{R_j})>$ to V_i.

- Step 5: $V_i \rightarrow R_j$: $<MAC>$

After receiving the message from R_j, V_i calculates temporary service key TSK_i= $H(a_3\|b_1\|AC_i\|0)$ by the received b_1, his own a_3 and AC_i, and then reveals (RID_j, b_2, T_{R_j}) by TSK_i. Next, V_i sends back MAC=$H(TSK_i^{"}$, b_2+1), where $TSK_i^{"}$=$H(a_3\|b_1\|AC_i\|1)$, for mutual authentication.

Finally, R_j verifies V_i by checking whether MAC is correct or not. If yes, V_i is convinced; otherwise, this session is dropped. In the end, both R_j and V_i take TSK_i=$H(a_3\|b_1\|AC_i\|k)$ for data encryption of the k-th session in the access service phase, where k=2, 3, 4, ..., and so on.

5.2.2 Comments on SECSPP

SECSPP gives a security solution for non-safety applications in VANETs. Both security and privacy issues were considered in the protocol design. However, the scalability issue is not addressed in SECSPP. As mentioned above, VANETs should be regarded as large scale networks. In SECSPP, only a single SP takes charge of checking the validity of authorized credential AC_i. This may lead to a bottleneck problem, or may introduce the threat of potential Distributed/Denial-of Service (D/DoS) attacks. In addition, SECSPP doesn't support differentiated service access control, which allows a variety of non-safety services with different privileges. It is believed that if non-safety applications try to achieve the success in VANETs, a sophisticated access control scheme [52] is required to meet a variety of users' demands.

In SECSPP, each SP_i needs two secret keys, SPK_{S_i} and SK_{S_i}. The former is used for non-interactive ID-based public key, and the latter is used for signing and decrypting the messages sent from vehicles or RSUs. This may cause inconvenience for SPs. In terms of

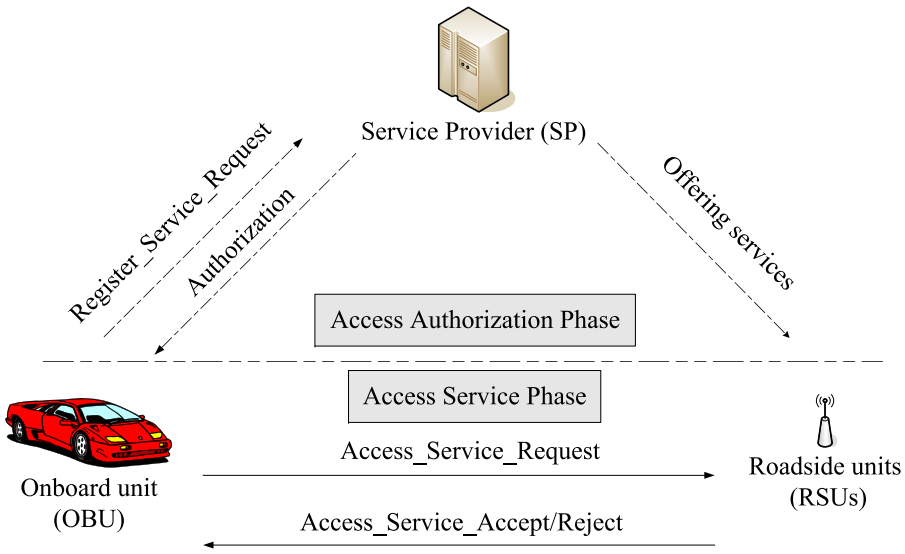


Figure 5.3.1: System architecture of a non-safety application

security, SECSPP adopted a conventional blind signature to prevent the vehicle's privacy from tracing by SPs. However, the conventional blind signature is not designed for access control. If SECSPP is adopted to provide the differentiated service access control, SECSPP could not withstand privilege elevation attacks [53], since SECSPP cannot examine whether the access privileges are valid or not. To deal with this weakness, we will first devise a novel attachable blind signature, and then develop a portable access control scheme based on the attachable blind signature. In addition, performance and scalability issues will be carefully examined in the design of our protocol.

5.3 The Portable privacy-preserving Authentication and Access Control Protocol (PAACP)

5.3.1 System Architecture

A system architecture of non-safety applications in VANETs is given in Figure 5.3.1. In general, a non-safety application of VANETs is composed of three types of entities, 1) onboard units (OBUs), 2) roadside units (RSUs), and 3) service providers (SPs). The

SPs are responsible for providing various non-safety services with the differentiated access privileges. For example, a travel company serves as the service provider to provide a travel guide service with two classes of customers, VIP and non-VIP customers. While a VIP customer uses the travel guide service, the travel company automatically pushes a bunch of coupons of local hotels or restaurants, which are only available for a VIP-exclusive service. In practice, an SP may deploy devices or databases in networks near RSUs for offering various non-safety services in a distributed fashion. Thus, the access of non-safety services can be fulfilled locally.

Initially, each OBU must send a *Register_ Service_ Request* message to the SP to request the authorization of the desired services in the access authorization phase. In the access service phase, when an OBU wants to access some services, the OBU delivers the *Access_ Service_ Request* message to its neighboring RSU. If the requesting OBU is authorized, then the neighboring RSU sends back the *Access_ Service_ Accept* message and allows the requesting OBU to access the desired services; otherwise, the OBU receives the *Access_ Service_ Reject* message without any access permission.

5.3.2 The Proposed Attachable Blind Signature

Generally, blind signatures could be implemented by different cryptosystems, such as RSA and ElGamal. We adopt RSA-based blind signature in the proposed blind signature scheme. First, we briefly introduce the conventional RSA-based blind signature. A user U_A blinds a message m with a random blind factor r and computes the blind document

$$BD = r^e m,$$

where e is the public key of the signer. The blind document is then sent to the signer. Once receiving BD, the signer signs BD by his/her private key d as

$$BD' = BD^d = rm^d.$$

Then the signer sends BD' back to U_A. Upon receiving BD', U_A unblinds BD' by the blind factor r to obtain the signer's signature

$$BD^{"} = m^d = BD'/r.$$

Finally, U_A confirms the integrity of $BD^{"}$ by checking

$$(BD^{"})^e = m.$$

In a conventional blind signature, the signer does nothing but signs the blind document BD sent from the user. Such a conventional blind signature is not designed for access control in origin. In terms of access control, the service provider (SP) plays the role of the signer and also confirms whether the requested access privileges for a user are legal. Since the blind document containing the requested access privileges is blinded by a random number r, it is infeasible for the SP to check whether the requested access privileges are legal. To ensure the genuineness of the requested access privileges, we propose an attachable blind signature as follows.

First, a user U_A chooses random blind factors r_1,r_2 and a, and then computes

$$BD_1 = (r_1)^e m^a (modN)$$

$$BD_2 = (r_2)^e m^{(1-a)} (modN).$$

Then, U_A sends BD_1 and BD_2 to the signer. Once receiving BD_1 and BD_2, the signer first attaches a message m' into BD_2 as

$$BD_2^{\#} = (r_2)^e m^{(1-a)} m'(modN),$$

and signs $BD_1, BD_2^{\#}$ by his/her own private key d as

$$BD_1^{'} = (BD_1)^d = r_1(m^a)^d (modN)$$

$$BD_2^{'} = (BD_2^{\#})^d = r_2(m^{(1-a)}m')^d (modN).$$

Then, the signer sends $BD_1^{'}$,$BD_2^{'}$ back to U_A. Upon receiving $BD_1^{'}$ and $BD_2^{'}$, U_A first unblinds two messages as

$$BD_1^U = BD_1^{'}/r_1 = (m^a)^d (modN)$$

$$BD_2^U = BD_2^{'}/r_2 = (m^{(1-a)}m')^d = (m^{(1-a)d})(m'^d)(modN)$$

and generates the signer's signature by

$$BD^{"} = BD_1^U \cdot BD_2^U = m^d \cdot m'^d (mod N).$$

Note that the proposed attachable blind signature scheme attaches a message m' into the signature and still keeps the privacy of user's message m. To withstand the privileges elevation attack, PAACP takes the advantage of m' to ensure the validity of m.

5.3.3 Portable Privacy-Preserving Authentication and Access Control Protocol (PAACP)

In this section, we propose a novel Portable privacy-preserving Authentication and Access Control Protocol (PAACP) for non-safety applications in VANETs. Since the stringent time requirement is regarded as an important property of VANETs [5, 6, 49], PAACP gets rid of the backend communication between roadside units and service providers. In PAACP, SPs do not involve in the access service phase. That is, the verification of vehicles and their access privileges can be accomplished in RSUs themselves. Thus, it is not required to take a long round trip of communication between RSUs and SPs for access request verifications. In the access authorization phase of PAACP, the SP authorizes the access privileges for a legitimate vehicle, and stores a service right list in a portable authorized credential carried by the vehicle. The portable authorized credential is protected using the proposed attachable blind signature to withstand privilege elevation attacks.

Another merit of PAACP is the support of differentiated access privileges for each service. A service may provide different access privileges to satisfy distinct requirements of the users. For this, the access privileges for the service i are represented by a bit string AR_i of k_i bits. Each bit of AR_i represents a distinct access privilege of the service i. In a travel guide service, for instance, we may use one bit to indicate the permission of viewing detailed maps, and one bit to indicate the permission of downloading coupons, and another bit to denote the capability of watching a particular video program. Therefore, k_i distinct access privileges can be specified in AR_i. Assume an SP provides n services with access privileges AR_i, $1 \leq i \leq n$. Suppose a vehicle V is granted to access m services, $1 \leq m \leq n$, with

index $\{SVID_1, SVID_2,\ldots,SVID_m\}$. Let $AR_j^{'}$, $1 \leq j \leq m$, be the granted value of AR_j for V. Then, the service right list SRL for V can be represented by a bit string

$$SRL = (SVID_1 \parallel AR_1^{'}) || (SVID_2 \parallel AR_2^{'}) \| \ldots \| (SVID_m \parallel AR_m^{'})$$

with length $\sum_{i=1}^{m}(logn + k_i)$. For example, we assume an SP provides 16 services and the travel guide is the 12-th service with three different access privileges: viewing maps, downloading coupons, watching videos, then n=16 and k_{12}=3 for AR_{12}. If V_i applies for the travel guide service with the access privileges of viewing maps and downloading coupons, then V will set SRL=(1100‖110) [53].

The proposed scheme consists of two phases: access authorization phase and access service phase, as illustrated in Figure 5.3.2 and 5.3.3. The notations of PAACP are summarized in Table 5.2.

According to the purchased services and granted access privileges, in the access authorization phase, a vehicle V_i creates a service right list $SRL_i^{V_i}$ in $AC_i^{V_i}$ and blinds $AC_i^{V_i}$ into blind documents $BD1_i$, $BD2_i$. To obtain the corresponding portable authorized credential for later use, V_i sends the blind documents with its certificate $Cert_i$ to the service provider S_t. After checking the validity of $Cert_i$, S_t generates the service right list $SRL_i^{S_t}$ based on the sold contract, stores $SRL_i^{S_t}$ in $AC_i^{S_t}$ and attaches $AC_i^{S_t}$ into blind documents $BD1_i$, $BD2_i$ based on the proposed attachable blind signature. Then, S_t delivers the blind documents back to V_i. At the end of the access authorization phase, V_i will obtain the portable authorized credential AC_i^* , where AC_i^* consists of both $AC_i^{V_i}$ and $AC_i^{S_t}$. AC_i^* is stored in V_i's tamper-proof device [5, 48, 49]. In the access service phase, V_i sends an *Access_ Service_ Request* to its neighboring RSU R_j, and then R_j verifies the authorized credential AC_i^* by itself without further communication with S_t. According to the access privileges stored in the authorized credential $AC_i^{S_t}$, R_j could decide whether V_i's request is accepted or not. Furthermore, R_j could detect whether V_i is launching a privilege elevation attack.

Table 5.2: Notations of the proposed scheme

V_i	the i-th vehicle
VID_i	The identification of the i-th vehicle
R_j	the j-th roadside unit
RID_j	The identification of the j-th roadside unit
S_t	the t-th service provider
SID_t	The identification of the t-th service provider
$SVID_i$	The identification of the i-th service
AR_i	The access privilege of $SVID_i$
(PK_{V_i}, SK_{V_i})	A public key and private key of vehicle V_i
(PK_{R_j}, SK_{R_j})	A public key and private key of roadside unit R_j
(PK_{S_t}, SK_{S_t})	A public key and private key of service provider S_t
$Cert_i$	The certificate of vehicle V_i
TSK	A temporary session key between the vehicle and roadside unit
a, RN_j	Random numbers, where j=1, 2.
AC_i	Authorized credential for vehicle V_i
AC^{S_t}, AC^{V_i}	Authorized credential made by S_t and V_i, respectively
AC_i^*	Portable authorized credential for vehicle V_i
SRL	The service right list
SRL^{S_t}, SRL^{V_i}	Service right list made by S_t, and V_i, respectively
$BD1$,$BD2$	The blind documents used in the proposed attachable blind signature
$E_{K_{AB}}\{\cdot\}$	The encryption function with shared key K_{AB}
$D_{K_{AB}}\{\cdot\}$	The decryption function with shared key K_{AB}
MAC	The message authentication code
$h(\ )$	A collision-free and public one-way hash function
σ_i	A signature signed by secret key SK_{V_i}
q	A large prime number
g	A generator of a finite cyclic group with order q.

We explain the details of each phase as follows.

5.3.4 Access Authorization Phase

- Step 1: $V_i \rightarrow S_t$: $< VID_i, \sigma_i, BD1_i, BD2_i >$

In the access authorization phase, according to the purchase receipt from the service provider S_t, a vehicle V_i creates its service right list $SRL_i^{V_i} = \{SVID_1 \| AR_1 \| .... \| SVID_k \| AR_k\}$, where $SVID_k$ denotes the index of the k-th service, and AR_k represents the granted access privileges of $SVID_k$. The service right list will be signed by S_t as part of an authorized credential. First, V_i chooses random numbers RN_1, RN_2 and a, and then sets $AC_i^{V_i} = \{SID_t \| T_{expired} \| SRL_i^{V_i}\}$. These random numbers are used as blind factors. Then, V_i computes blind documents

$$BD1_i = (RN_1)^{PK_{S_t}} \cdot (AC_i^{V_i})^a (modN)$$

$$BD2_i = (RN_2)^{PK_{S_t}} \cdot (AC_i^{V_i})^{1-a} (modN).$$

Finally, V_i sends its identity VID_i, signature $\sigma_i = \{BD1_i, BD2_i\}^{SK_{V_i}}$, and the blinded documents $BD1_i$, $BD2_i$ to S_t.

- Step 2: $S_t \rightarrow V_i$: $< BD1_i^{'}, BD2_i^{'} >$

Upon receiving message $< VID_i, \sigma_i, BD1_i, BD2_i >$ sent from V_i, S_t first confirms whether the σ_i is valid by V_i's public key. If valid, V_i is successfully authenticated; otherwise, this session is dropped. S_t then generates the authorized credential $AC_i^{S_t} = \{SID_t \| T_{expired} \| SRL_i^{S_t}\}$ according to the selling contract for V_i and attaches it into $BD2_i^{\#}$ as

$$BD2_i^{\#} = BD2_i \cdot AC_i^{S_t} = ({RN_2}^{PK_{S_t}} \cdot (AC_i^{V_i})^{1-a} \cdot AC_i^{S_t}) (modN).$$

Then, S_t signs them as follows.

$$BD1_i^{'} = {BD1_i}^{SK_{S_t}} = (RN_1)^{PK_{S_t}} \cdot (AC_i^{V_i})^{a^{SK_{S_t}}} = (RN_1) \cdot (AC_i^{V_i})^{a^{SK_{S_t}}}$$

$$BD2_i^{'} = BD2_i^{\#^{SK_{S_t}}} = (RN_2)^{PK_{S_t}} \cdot (AC_i^{V_i})^{1-a} \cdot AC_i^{S_t{}^{SK_{S_t}}} = (RN_2) \cdot (AC_i^{V_i})^{1-a} \cdot AC_i^{S_t{}^{SK_{S_t}}}$$

Next, $BD1_i^{'}$, $BD2_i^{'}$ are sent back to V_i. After obtaining $<BD1_i^{'}, BD2_i^{'}>$ from S_t, V_i unblinds them as follows.

$$BD1_i^U = BD1_i^{'}/RN_1 = (AC_i^{V_i})^{a^{SK_{S_t}}}$$

$$BD2_i^U = BD2_i^{'}/RN_2 = (AC_i^{V_i})^{1-a} \cdot AC_i^{S_t{}^{SK_{S_t}}}$$

In order to get the portable authorized credential $AC_i^* = \{AC_i^{V_i} \cdot AC_i^{S_t}\}^{SK_{S_t}}$, V_i computes

$$BD1_i^U \cdot BD2_i^U = (AC_i^{V_i})^{a^{SK_{S_t}}} \cdot (AC_i^{V_i})^{1-a} \cdot AC_i^{S_t{}^{SK_{S_t}}} = AC_i^{V_i} \cdot AC_i^{S_t{}^{SK_{S_t}}}.$$

To confirm the AC_i^* is certified, V_i could verify the correctness of AC_i^* by checking whether $\{AC_i^*\}^{PK_{S_t}}$ is equal to $AC_i^{V_i} \cdot AC_i^{S_t}$.[2] If it holds, V_i keeps AC_i^* for the subsequent service requests; otherwise, V_i will stop this session. Note that, we assume V_i could protect AC_i^* in secret by tamper-proof device after obtaining AC_i^*.

5.3.5 Access Service Phase

- Step 1: $V_i \rightarrow R_j$: $<Access\_Service\_Request, \{SVID, Y_V, AC_i^*\}^{PK_{R_j}}>$

In the access service phase, when a vehicle V_i wants to access the desired services from its neighboring roadside unit R_j, V_i will transmit an $Access\_Service\_Request$ with $\{SVID, Y_V, AC_i^*\}^{PK_{R_j}}$, where $SVID$ is the identification of the desired services, and $Y_V = g^x \, mod \, q$, where x is a random number in $\mathbb{Z}_q^*$, to R_j.

- Step 2: $R_j \rightarrow V_i$:$<Y_R, E_{TSK^0}\{Y_V+1, Access\_Permission\}>$

[2]Note that if both V_i and S_t are legal, $AC_i^{V_i}$ and $AC_i^{S_t}$ should be the same, which means V_i or R_j could confirm whether $AC_i^{V_i} \cdot AC_i^{S_t}$ is expected or not.

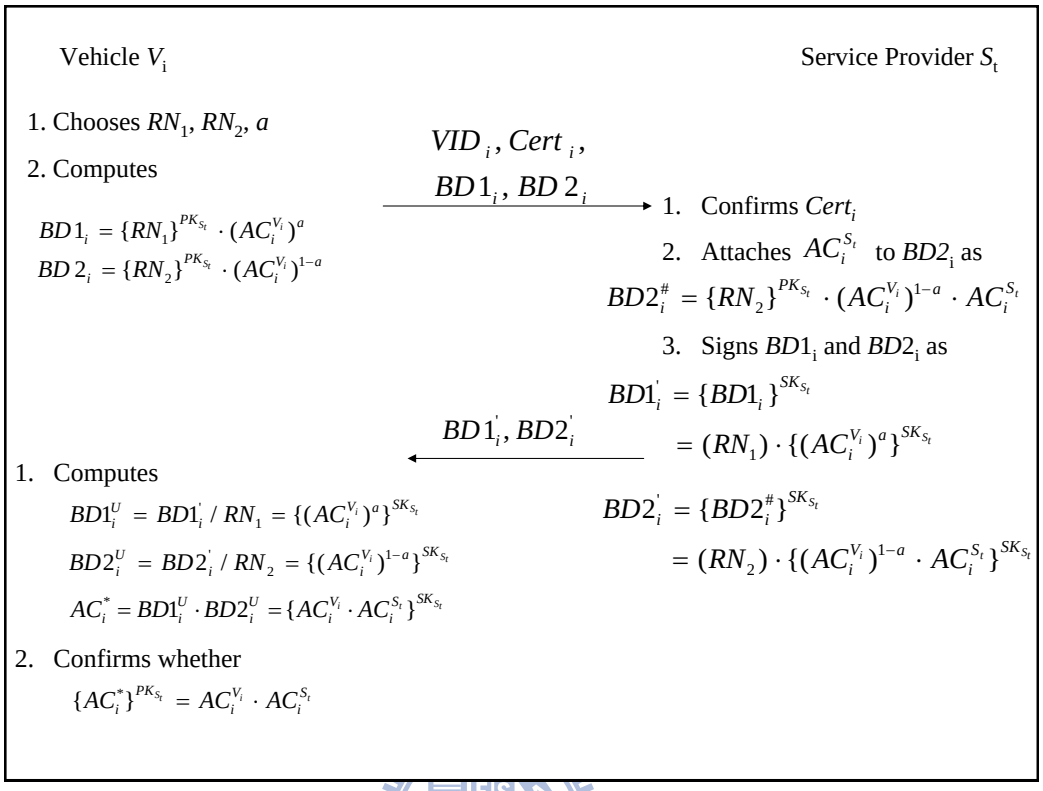


Figure 5.3.2: Access authorization phase of the proposed scheme

Upon receiving $\{SVID, Y_V, AC_i^*\}^{PK_{R_j}}$, R_j decrypts it by his own private key SK_{R_j} to acquire $(SVID, Y_V, AC_i^*)$. First, R_j calculates

$$AC_i^{S_t} = (AC_i^{*PK_{S_t}})^{1/2}$$

to extract the access credential $AC_i^{S_t}$, which is authorized by S_t. Then, R_j examines whether SID_t as well as $SVID$ is included in $AC_i^{S_t}$, and checks the validity of the authorized credential by $T_{expired}$. If the verification succeeds, AC_i^* is legitimate and V_i is authorized; otherwise, R_j terminates this session. After AC_i^* is verified, R_j calculates

$$Y_R = g^y \bmod q,$$

where y is a random number in $\mathbb{Z}_q^*$, and generates a temporary session key

$$TSK^0 = h(AC_i^*, (Y_V)^y \bmod q, 0)$$

for protecting the subsequent communications. Finally, R_j delivers $<Y_R$, $E_{TSK^0}\{Y_V+1$, $Access\_Permission\}>$ to V_i.

- Step 3: $V_i \rightarrow R_j$: $<E_{TSK^1}\{Auth\_Ack\}, MAC>$

After receiving $<Y_R$, $E_{TSK^0}\{Y_V+1$, $Access\_Permission\}>$, V_i computes a temporary session key

$$TSK^0 = h(AC_i^*, (Y_R)^x \, mod \, q, 0)$$

and decrypts $E_{TSK^0}\{Y_V+1$, $Access\_Permission\}$ using TSK^0 to check the validity of Y_V+1. If valid, R_j is successfully authenticated; otherwise, V_i ceases this connection. Then, V_i generates an $Auth\_Ack$ encrypted by

$$TSK^1 = h(AC_i^*, (Y_R)^x \, mod \, q, 1)$$

and computes the message authentication code

$$MAC = (TSK^0, E_{TSK^1}Auth\_Ack).$$

Finally, V_i sends $<E_{TSK^1}\{Auth\_Ack\}$, $MAC>$ to R_j.

Upon receiving the message, R_j verifies the MAC to ensure the integrity, and calculates

$$TSK^1 = h(AC_i^*, (Y_V)^y \, mod \, q, 1)$$

to decrypt $E_{TSK^1}\{Auth\_Ack\}$. If R_j could recognize $Auth\_Ack$, it is implied that V_i indeed holds the corresponding TSK^1. Finally, the subsequent communications can be encrypted by the session key TSK^k, where

$$TSK^k = h(AC_i^*, (Y_V)^y \, mod \, q, k).$$

5.4 Security and Correctness Analysis

5.4.1 Security Properties

Based on the security of asymmetric and symmetric cryptosystems, PAACP preserves several security properties, as discussed below.

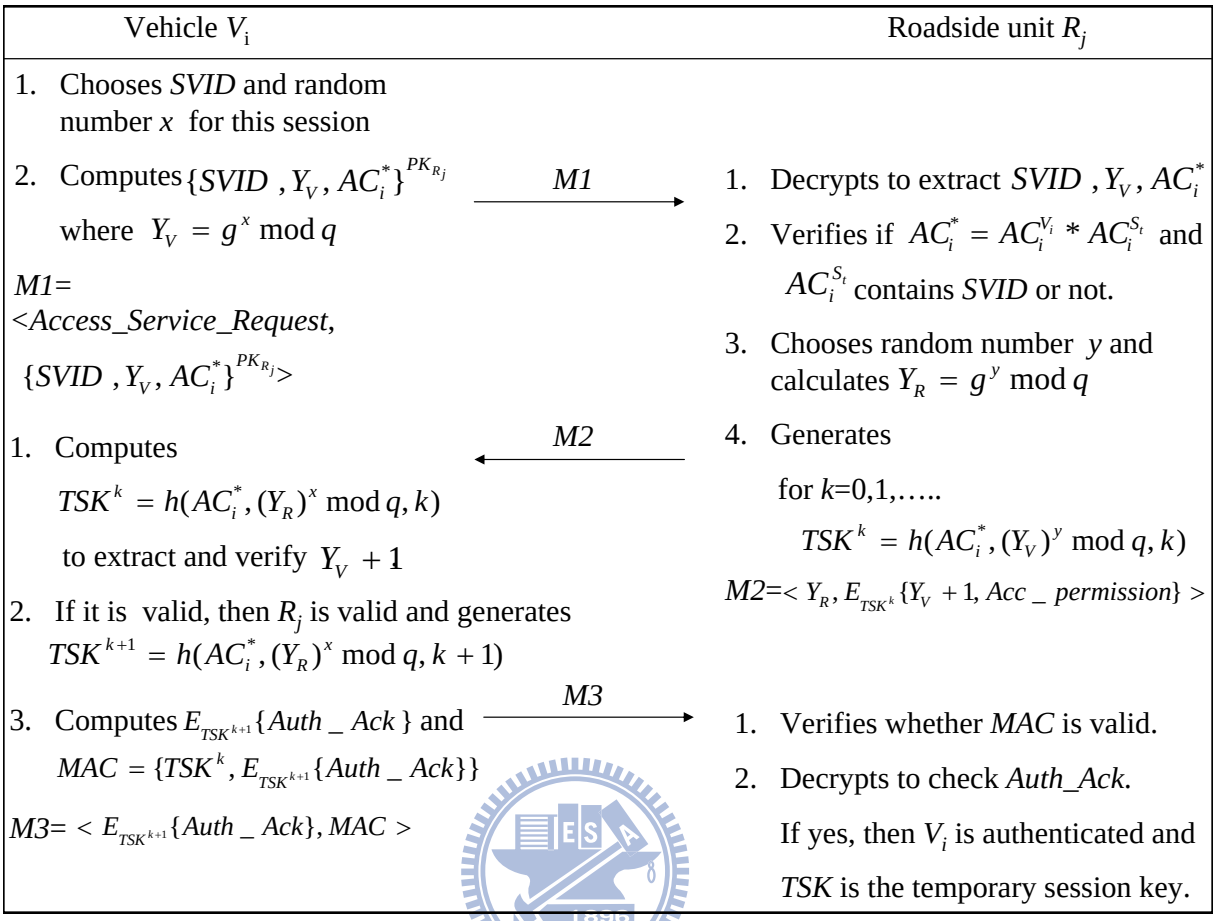


Figure 5.3.3: Access service phase of the proposed scheme

5.4.1.1 Mutual Authentication

In PAACP, vehicle V_i and roadside unit R_j are mutually authenticated based on the secret authorized credential AC_i^* and the public key cryptosystem. Only an authorized V_i could own AC_i^*, and only legitimate R_j has the capability of decrypting messages to extract Y_V. Mutual authentication is an essential property to prevent malicious attacks from outsiders.

5.4.1.2 Context Privacy

Based on the proposed attachable blind signature, no one could comprehend the access privileges in $AC_i^{V_i}$. Note that even if service provider S_t could realize V_i's access privileges in the access authorization phase, the non-linkability discussed in next subsection is also guaranteed. In the access service phase, all messages are well protected by asymmetric and symmetric cryptographic primitives without disclosing any information to outsiders. On

the other hand, although the roadside unit R_j can confirm the validity of the authorized credential AC_i^* and the desired services $SVID$, R_j cannot realize who is accessing those services.

5.4.1.3 Non-linkability

In general, the non-linkability means both insiders and outsiders could neither realize any session to a particular user nor link any two different sessions to the same user. First, PAACP ensures that outsiders cannot attain any information in the communications between V_i and R_j . Therefore, the non-linkability for outsiders is guaranteed under the security of asymmetric and symmetric cryptosystems. On the other hand, service provider S_t cannot link any sessions to a particular user since S_t is not involved in the access service phase. Moreover, even if S_t obtains the authorized credential AC_i^*, the non-linkability is still ensured by the proposed attachable blind signature since S_t cannot link this AC_i^* to the exact vehicle, unless the service right list itself is distinct for a certain vehicle. It is possible that R_j could link the authorized credential AC_i^* to the same vehicle, but R_j cannot derive any additional information about the vehicle.

5.4.1.4 Data Traffic Protection

After the execution of PAACP, all messages between V_i and R_j are encrypted by the session key TSK. Under the security of symmetric cryptographic primitive such as AES, the data confidentiality and integrity are guaranteed as well.

5.4.1.5 Differentiated Service Access Control

Different from the previous work [52] adopting several public/private key pairs to achieve the differentiated service access control, PAACP only requires a single public/private key pair and uses an SRL [53] to encode the access privileges of each services. As a result, PAACP also keeps the privacy of the service request in the access service phase.

Table 5.3: The comparison of security features

	Ours	SECSPP [10]	[49]
Mutual authentication	Yes	Yes	Yes
Context privacy	Yes	Yes	Yes
Session key agreement	Yes	Partially yes[4]	Partially yes[5]
Differentiated service access control	Yes	No	No
Privilege elevation attack resistance	Yes	N/A	N/A
Scalability	Fully distributed	Bottleneck at service provider	N/A
Formal correctness proof	Yes	No	No

[4]In SECSPP, the session key TSK is determined by V and S, not V and R.

[5]In Wang et al's scheme, the session key TSK is built for inter-vehicle communication (IVC), not V and R.

5.4.1.6 Forward Secrecy

Different from the previous works [10, 52], PAACP applies the concept of Diffie-Hellman exchange protocol using $Y_V{=}g^x \; modq$ and $Y_R{=}g^y \; modq$ to establish the session key $TSK^i{=}h(AC_i^*, g^{xy} = (Y_V)^y = (Y_R)^x, i)$. This implies that PAACP preserves the forward secrecy property even though a long-term secret key is compromised.

5.5 Discussion

5.5.1 Comparsion

In this section, we compare PAACP with the related works [10, 49] in terms of security properties and performance evaluation. First, we compare the security features of PAACP with SECSPP and [49], which are typical authentication schemes for non-safety applications in VANETs. Table 5.3 lists important security properties in VANETs. The comparison shows that PAACP provides more merits, including differentiated service access control, privilege elevation attack resistance, and better scalability.

5.5.2 Performance Evaluation

Next, we evaluate the performance of SECSPP and PAACP in Table 5.4. For time complexity estimation, we define some computational parameters as follows:

T_{Asym} : the time for the asymmetric encryptions/decryptions.

T_{sym}: the time for the symmetric encryptions/decryptions.

$T_{ID_{exp}}$: the time for the modular exponentiation of the ID-based cryptography.

T_{hash}:the time for the one-way hash function operation.

T_{xor}: the time for the XOR operation.

Based on the computation method in [10] and [49], PAACP takes 2.0885 seconds to compute the necessary operations and SECSPP spends 2.0895 seconds in the authorization phase. In the access service phase, the verification time $T_{verification}$ of PAACP is 1.5839 seconds/time and that of SECSPP is 2.613 seconds/time. Note that the time spent in the access service phase is the major concern in terms of performance, since the access service phase will be executed frequently, whereas the authorization phase is executed only once. In addition to the required computation time in the access service phase, the overall elapsed time can be evaluated by the communication rounds needed and the waiting time for each vehicle when there are a number of service requests simultaneously. In general, the service provider is far away from RSUs, but vehicles are in the neighborhood of RSUs. Let $T_{trans-delay}$ be the transmission delay in seconds to deliver a message from a vehicle, forwarded by an RSU, to the SP. It is reasonable to assume $1 < T_{trans-delay} < 2$. The transmission delay in seconds to deliver a message from a vehicle to its neighboring RSU is less than 0.01 seconds [1], which can be neglected. Considering the scalability issue, we further assume that n vehicles in the VANET request the services of the same SP at the same time and the locations where these service requests are invoked are uniformly distributed within m RSUs [8]. In SECSPP, the average waiting time $T_{waiting}$ for a requesting vehicle can be estimated as

$$T_{waiting_{SECSPP}} = 2T_{trans-delay} + (n+1)/2 * T_{verification},$$

The waiting time consists of round-trip transmission delay and the time spent in the SP for verification. Since thre are n requests pending for verification, the average time spent in SP will be $(n+1)/2 * T_{verification}$. On the other hand, in PAACP, the average waiting time $T_{waiting}$ for a requesting vehicle can be estimated as

Table 5.4: The comparison of efficiency

	Ours	SECSPP [10]
Authorization phase	4 T_{Asym} + 1 T_{hash}	2 T_{Asym} + 2 T_{IDexp}+ 3 T_{hash}+ 4 T_{xor}
Access service phase	3 T_{Asym}+ 2 T_{sym}+ 1 T_{hash}	3 T_{Asym} +2 T_{IDexp}+ 6 T_{hash}+ 5 T_{xor}
Computation time (s)	Authorization phase ≈ 2.0885 (s)	Authorization phase ≈ 2.0895 (s)
	Access service phase ≈ 1.5839 (s)	Access service phase ≈ 2.613 (s)
Communication rounds	2+3	2+5

$$T_{waiting_{PAACP}} = \begin{cases} (n/m+1)/2 * T_{verification}, & \text{if } n > m \\ T_{verification}, & \text{otherwise} \end{cases}$$

In a uniform distribution of locations, the avergae number of requests pending in each RSU will be n/m. Therefore, the average time spent for request verification in a RSU is $(n/m+1)/2 * T_{verification}$. Figure 5.5.1 shows the average waiting time for a service request as n increases with different values of m (10, 30, and 50). As Figure 5.5.1 (a), (b) and (c) shows, when 100 vehicles are requesting the desired services, the average waiting time to finish the authentication in SECSPP is 134 seconds. As for PAACP, the waiting times for m=10, 30, and 50 take about 10, 5, and 3 seconds, respectively. The waiting time for PAACP is short since the verifications of access requests can be performed locally because of the distributed nature of PAACP. Moreover, the more RSUs are installed, the less waiting time in PAACP is required. In terms of communication rounds, PAACP eliminates the transmission overhead between RSUs and SPs. Hence, the total number of communication rounds required in PAACP is lower than that of SECSPP, as shown in Figure 5.5.2. Obviously, the number of communication rounds of PAACP is 60 % fewer thant that of SECSPP. In summary, PAACP outperforms SECSPP significantly.

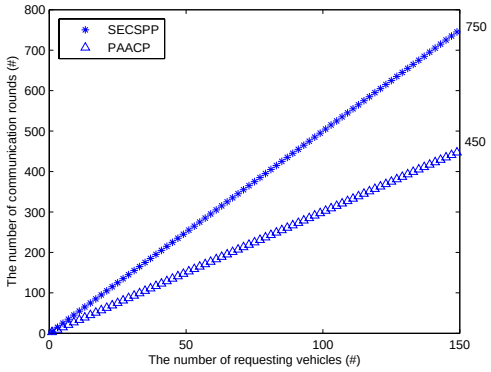
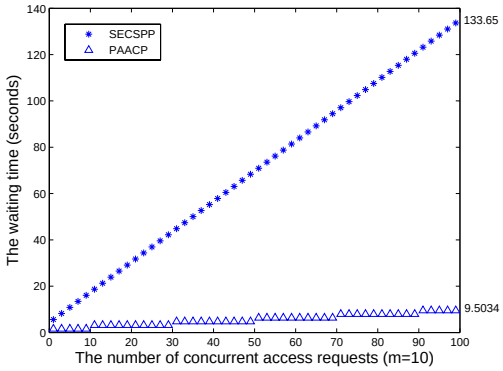
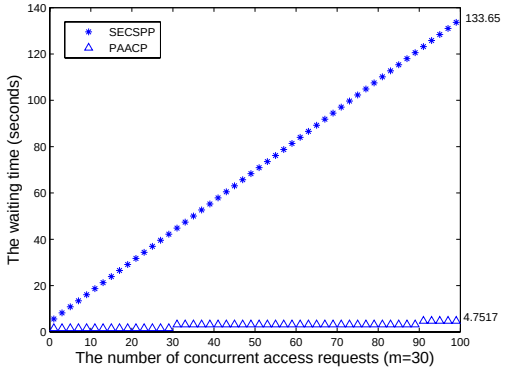


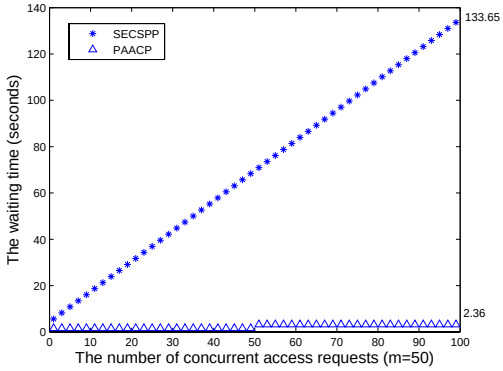
Figure 5.5.2: Communication rounds v.s. requesting vehicles



(a) The average waiting time v.s. the number of concurrent access requests (m=10)



(b) The average waiting time v.s. the number of concurrent access requests (m=30)



(c) The average waiting time v.s. the number of concurrent access requests (m=50)

Figure 5.5.1: Average wating time v.s.concurrent access requests

Chapter 6

Conclusions and Future Work

Recently, emergency management in intelligent transportation systems (ITSs) has attracted considerable attention. Current security schemes over VANETs will thus become candidates for use in future ITSs. Most of these approaches focus on the security and privacy of message verification. In our dissertation, we first have proposed an *A*ttribute-*B*ased *A*ccess *C*ontrol *S*ystem (ABACS) for emergency services over VANETs. The attributed-based secure multicast scheme adopted in ABACS can efficiently find and select emergency vehicles over VANETs. Our analysis shows that both security and better efficiency can be realized using ABACS. In ABACS, we have defined several messages for use in an emergency service. With regard to an emergency service in the real world, it is noted that different definitions of data fields in these messages may be required. Nevertheless, our theoretical approach can be regarded as the very first attempt to define a secure framework for providing emergency services over VANETs. Second, we have also proposed a novel anonymous batch authenticated and key agreement scheme (ABAKA) for value-added services in VANETs. With ABAKA, a service provider can simultaneously authenticate multiple requests and establish different session keys with vehicles. To deal with the invalid request problem, a detection algorithm has also been proposed. Moreover, the efficiency and practicality to the real-world applications have been verified by the simulation analysis. Last, we have proposed a *P*ortable privacy-preserving *A*uthentication and *A*ccess *C*ontrol *P*rotocol (PAACP) for one kind of non-safety applications in VANETs. Considering the stringent time requirement in VANETs, we devised a portable access control protocol to get rid of the involvement of

service providers in the access service phase. Due to the portability of authorized service right lists, roadside units can verify the validity of access privileges without the aid of service providers. Moreover, we proposed an attachable blind signature to keep the privacy of the requested services and to withstand the privilege elevation attack. The performance evaluations also show that PAACP is efficient and suitable for large scale VANETs.

Our future work will be on the investigation of more emergency scenarios, e.g., rescues for mass disasters, optimal mission assignment for multiple emergency events, and emergency services for different VANET configurations. In addition, the features of VANETs, such as the mobility model and predicable routing, could be taken into consideration to to gain more efficiency. In V2V communications, how to ensure the security issues without the help of RSUs should be addressed as well.

Bibliography

[1] O. A. http://grouper.ieee.org/groups/scc32/dsrc/index.html, *Dedicated Short Range Communications (DSRC)*.

[2] L. Zhang, Q. Wu, A. Solanas, and J. Domingo-Ferrer, "A scalable robust authentication protocol for secure vehicular communications," *IEEE Trans. on Vehicular Technology*, vol. 59, no. 4, pp. 1606–1617, 2010.

[3] X. s. H. Zhu, R. Lu and X. Lin, "Security in service-oriented vehicular networks," *IEEE Wireless Communications*, pp. 16–22, 2009.

[4] X. Lin, X. Sun, P. H. Ho, and X. Shen, "Gsis: a secure and privacy-preserving protocol for vehicular communications," *IEEE Transaction on Vehicular Technology*, vol. 56, No. 6, pp. 3442–3456, 2007.

[5] C. Zhang, R. Lu, X. Lin, P. H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proc. IEEE INFOCOM*, 2008, pp. 816–824.

[6] C. Zhang, X. Lin, R. Lu, P. H. Ho, and X. Shen, "An efficient message authentication scheme for vehicular communications," *IEEE Transaction on Vehicular Technology*, vol. 57, No. 6, pp. 3357–3368, 2008.

[7] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications," in *Proc. IEEE INFOCOM*, 2008, pp. 1229–1237.

[8] X. Lin, X. Sun, X. Wang, C. Zhang, P.-H. Ho, and X. Shen, "Tsvc: timed efficient and secure vehicular communications with privacy preserving," *IEEE Transaction on Wireless Communications*, vol. 7, No. 12, pp. 4987–4998, 2008.

[9] *IEEE Trial-Use Standard for Wireless Access in Vehicular Environment-Security Services for Applications and Management Messges.* IEEE std. 1609.2-2006, Jul. 2006.

[10] C.-T. Li, M.-S. Hwang, and Y.-P. Chu, "A secure and efficient communication scheme with authenticated key establishment and privacy perserving for vehicular ad hoc networks," *Computer Communications*, vol. 31, pp. 2803–2814, 2008.

[11] M. S. P.-H. H. H. Zhu, X. Lin and X. Shen, "Ppab: A privacy-preserving authentication and billing architecture for metropolitan area sharing networks," *IEEE Trans. on Vehicular Technology*, vol. 58, no. 5, pp. 2529–2543, 2009.

[12] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, pp. 39–68, 2007.

[13] H. Zhu, X. Lin, R. Lu, P. H. Ho, and X. Shen, "Aema: An aggregated emergency message authentication scheme for enhancing the security of vehicular ad hoc networks," in *Proc. ICC*, 2008, pp. 1436–1440.

[14] "Vehicle safety communications project," U.S. Department of Transportation, National Highway Traffic Safety Administration, Tech. Rep., 2006.

[15] M. Dikaiakos, A. Florides, T. Nadeem, and L. Iftode, "Location-aware services over vehicular ad-hoc networks using car-to-car communication," *IEEE Journal on Selected Areas in Communications*, vol. 25, No. 8, pp. 1590–1602, 2007.

[16] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux, "Eviction of misbehaving and faulty nodes in vehicular networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, No.8, pp. 1557–1568, 2007.

[17] Y. Bi, H. Zhao, and X. Shen, "A directional broadcast protocol for emergency message exchange in inter-vehicle communications," in *IEEE International Conference on Communication (ICC)*, 2009, pp. 1–5.

[18] J. Peng and L. Cheng, "A distributed mac scheme for emergency message dissemination in vehicular ad hoc networks," *IEEE Transaction on Vehicular Technology*, vol. 56, No. 6, pp. 3300–3308, 2007.

[19] E. V. de Velde and C. Blondia;, "Adaptive react protocol for emergency applications in vehicular networks," in *IEEE Conference on Local Computer Networks (LCN)*, 2007, pp. 613–619.

[20] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of Eurocrypt*, LNCS, Ed., vol. 3494. Springer-Verlag, 2005, pp. 457–473.

[21] A. Shamir, "How to share a secret," *Communication of ACM*, vol. 22, No. 11, pp. 612–613, 1979.

[22] P. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps," in *Proceedings of Asiacrypt*, 2005, pp. 515–532.

[23] J. Zhao, Y. Zhang, and G. Cao, "Data pouring and buffering on the road: a new data dissemination paradigm for vehicular ad hoc networks," *IEEE Transaction on Vehicular Technology*, vol. 56, No. 6, pp. 3266–3277, 2007.

[24] M. Nekovee, "Modeling the spred of worm epidemics in vehicular ad hoc networks," in *Proceeding of Veh. Tech. Conf.*, 2006, pp. 115–124.

[25] J. E. Khoury and A. Hobeika, "Incorporating uncertainty into the estimation of the passing sight distance requirements," *Computer-Aided Civil and Infrastructure Engineering*, vol. 22, no. 5, pp. 347–357, 2007.

[26] S. Yu, K. Ren, and W. Lou, "Fdac: Toward fine-grained distributed data access control in wireless sensor networks," in *Proc. IEEE INFOCOM*, 2009, pp. 816–824.

[27] A. Wasef, Y. Jiang, and X. Shen, "Dcs: An efficient distributed certificate services scheme for vehicular networks," *IEEE Transaction on Vehicular Technology*, vol. 59, No. 2, pp. 533–549, 2010.

[28] J. Yin, T. ElBatt, G. Yeung, B. Ryu, S. Habermas, H. Krishnan, and T. Talty, "Performance evaluation of safety application over dsrc vehicular ad hoc networks," in *ACM Int. Workshop Vehicular Ad hoc Network*, 2004, pp. 1–9.

[29] O. A. http://nsnam.isi.edu/nsnam/index.php, *The Network Simulator-ns-2*.

[30] M. Piorkowski, M. Raya, A. L. Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux, "Trans: Realistic joint traffic and network simulator for vanets," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 12, No. 1, 2008.

[31] S. Lee, G. Pan, J. Park, M. Gerla, and S. Lu, "Secure incentives for commercial ad dissemination in vehicular networks," in *Proc. ACM Int. Symp. MobiHoc*, 2007, pp. 150–159.

[32] M. Raya, P. Papadimitratos, and J. P. Hubaux, "Securing vehicular communications," *IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications*, vol. 13, No. 5, pp. 8–15, 2006.

[33] B. C. Neuman and T. Ts'o, "Kerberos: An authentication service for computer networks," *IEEE Communications*, vol. 32, No. 9, pp. 33–38, 1994.

[34] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*. LNCS, Springer-Verlag, 2004.

[35] F. Li, X. Xin, and Y. Hu, "Identity-based broadcast signcryption," *Computer Standard and Interfaces*, vol. 30, pp. 89–94, 2008.

[36] J.-H. Yang and C.-C. Chang, "An id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *Computer & Security*, vol. 28, pp. 138–143, 2009.

[37] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions of elliptic curve traces for fr-reduction," *IEICE Transactions on Fundamentals*, vol. E84-A, pp. 1234–1239, 2001.

[38] J. Camenisch, S. Hohenberger, and M. O. Pedersen, "Batch verification of short signatures," in *Proceedings of EUROCRYPT*, 2007, pp. 246–263.

[39] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, 2001.

[40] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proceedings of Eurocrypt*, vol. 2656, 2003, pp. 416–432.

[41] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Proceedings of Asiacrypt*, vol. 2248, 2001, pp. 514–532.

[42] M. Scott, "Efficient implementation of cryptographic pairings," [on-line available] http://ecrypt-ss07.rhul.ac.uk/slides/thursday/mscottsamos07.pdf.

[43] S. Yousefi, M. Mousavi, and M. Fathy, "Vehicular ad hoc networks (vanets): challenge and persepectives," in *International Conference on ITS telecommunications*, 2006, pp. 761–766.

[44] J. T. Isaac, J. S. Camara, S. Zeadally, and J. T. Marquez, "A secure vehicle-to-roadside communication payment protocol in vehicular ad hoc networks," *Computer Communications*, vol. 31, pp. 2478–2484, 2008.

[45] "Kvh industries inc. [online]. available: http://www.kvh.com/," 2006.

[46] "Msn tv. [online]. available: http://www.msntv.com/," 2006.

[47] Car-2-Car., "[online]. available: http://www.car-2-car.org," 2007.

[48] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *SASN*, 2005, pp. 11–21.

[49] N.-W. Wang, Y.-M. Huang, and W.-M. Chen, "A novel secure communication scheme in vehicular ad hoc networks," *Computer Communications*, vol. 31, pp. 2827–2837, 2008.

[50] X. Lin, R. Lu, C. Zhang, H. Zhu, P.-H. Ho, and X. Shen, "Secuity in vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 88–95, 2008.

[51] Y. Toor, P. Muhlethaler, and A. Laouiti, "Vehicle ad hoc networks: applications and related technical issues," *IEEE Communications Surveys & Tutorials*, vol. 10, pp. 74–87, 2008.

[52] K. Ren, W. Lout, K. Kim, and R. Deng, "A novel privacy preserving authentication and access control scheme for pervasive computing environments," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 4, pp. 1373–1384, 2006.

[53] Y.-C. Chen and L.-Y. Yeh, "An efficient authentication and access control scheme using smart cards," in *Proceedings of International Conference on Parallel and Distributed Systems (ICPADS)*, 2005, pp. 78–82.

Vita

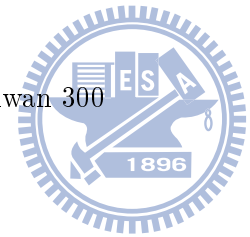
Lo-Yao Yeh

Department of Computer Science

National Chiao Tung University

1001 Ta Hsueh Road, Hsinchu, Taiwan 300

Email: lyyeh@cs.nctu.edu.tw



Education

Ph.D.: Computer Science, National Chiao Tung University (2005.9 to 2010.12)

M.S.: Information Management, National Chi Nan University (2003.9 to 2005.6)

B.S.: Information Management, Da Yeh University (1999.9 to 2003.6)

Publications

- Journal Papers

1. Lo-Yao Yeh, Yen-Cheng Chen, and Jiun-Long Huang,"ABACS: An Attribute-Based Access Control System for Emergency Services over Vehicular Ad Hoc Networks," to appear in *IEEE Journal on Selected Areas in Communications - special issue on Vehicular Communications and Networks.* (SCI)
2. Lo-Yao Yeh, Jiun-Long Huang, Hung-Yu Chien, "ABAKA: An anonymous batch authenticated and key agreement scheme for value-added service in vehicular ad hoc network," to appear in *IEEE Trans. on Vehicular Technology.* (SCI, EI)
3. Lo-Yao Yeh, Yen-Cheng Chen, and Jiun-Long Huang," PAACP: A Portable Privacy-Preserving Authentication and Access Control Protocol in Vehicular Ad hoc Networks," to appear in *Computer Communications.* (SCI)
4. Lo-Yao Yeh , Chen-Che Huang, Cheng-En Wu, and Jiun-Long Huang," ALM: An Adaptive Location Management Scheme for Approximate Location Queries in Wireless Sensor Networks," *Computer Communications*, Vol. 33, No. 16, pp. 1937-1948. (SCI)
5. Yen-Cheng Chen, Shu-Chuan Chuang, Lo-Yao Yeh, and Jiun-Long Huang,"A Practical Authentication Protocol with Anonymity for Wireless Access Networks," to appear in *Wireless Communications and Mobile Computing.* (SCI)
6. Woei-Jiunn Tsaur and Lo-Yao Yeh ,"A Novel Mobile Agent Authentication Scheme for Multi-host Environments Using Self-certified Pairing-based Public Key Cryptosystem," to appear in *International Journal of Innovative Computing, Information and Control*, Vol. 6, No.2, 2010. (SCI)
7. Yen-Cheng Chen and Lo-Yao Yeh, "A Portable Integrated Authentication and Access Control Scheme for Distributed Embedded Systems," *International Journal of Innovative Computing, Information and Control* , Vol. 6, No.2, 2010. (SCI)
8. Yen-Cheng Chen and Lo-Yao Yeh,"An Efficient Nonce-Based Authentication Scheme with Key Agreement," *Applied Mathematics and Computation*, Vol. 169, pp. 982-994, 2005. (SCI)

- Conference Papers

1. G.-H. Ye, L.-Y. Yeh and J.-L. Huang, "Hierarchical Role-based Data Dissemination for Large-Scale Wireless Sensor Networks with Mobile Sinks," *IEEE PerCom Workshops* , 2010.
2. Y. L. Huang, J. D. Tygar, H. Y. Lin, L. Y. Yeh, H. Y. Tsai, K. Sklower, S. P. Shieh, C. C. Wu, P. H. Lu, S. Y. Chien,"SWOON: A Testbed for Secure Wireless Overlay Networks," *USENIX Workshop on Cyber Security Experimentation and Test*, 2008.
3. Yen-Cheng Chen, and Lo-Yao Yeh, "An Efficient Authentication and Access Control Scheme Using Smart Cards," *IEEE ICPADS Workshops*, Fukuoka, Japan, July 20 - 22, 2005.