Master Thesis

Secure and Privacy-Preserving Broadcast Authentication for IVC

Author:
Liting Huang
s1017241
younggery@gmail.com

Graduation Committee:
dr. F. Kargl
dr.ir. G.J. Heijenk
dr. J.Y. Petit

Distributed and Embedded Security Group,
Faculty of Electrical Engineering,
Mathematics and Computer Science

UNIVERSITEIT TWENTE.

July 2, 2012

Acknowledgements

I want to thank my supervisor Frank Kargl for giving me valuable guidance and instructions on the thesis. And I also want to thank Jonathan Petit, who is always helpful in the details of my project. I would like to thank Geert Heijenk for spending his time in reading my thesis. I want to give special thanks to my friend Arthur, who helped check my spelling mistakes and polish the language.

I am grateful for my parents, my roommates, and other friends who often asked about the progress of my thesis and who gave their suggestions on my life and study. They also delighted me on the way of studying.

Abstract

Vehicle-to-Vehicle(V2V) communication is a part of the future vehicular network. As the location information of vehicles is broadcasted frequently, there is a demand on privacy protection on this information. In this thesis we defined the requirements on privacy-protection broadcast authentication schemes for V2V communication. We analyzed the existing authentication schemes according to the requirements. But the major contribution of this thesis is that we devised an authentication scheme CLIBA on the messages of vehicles, which is based on the CL-Idemix protocol suite. The scheme realizes attribute authentication to prevent privacy leakage of vehicles. We also evaluated CLIBA according to the requirements. It shows that CLIBA fulfills most of the requirements except that the performance is not quite satisfactory compared to the strict efficiency requirement of V2V communication.

Contents

Chapter 1

Introduction & Motivation

1.1 Vehicular Networks

As a part of ubiquitous computing, "road automation" has been under discussion and research for many years. Ever since the basic concept of "road automation" was introduced in 1939, the investigation in wireless communication around vehicles has changed its focuses alongside its development. Route-guidance systems, tolling systems and automatic driving used to be the hot topics[20]. Products for those systems have already been developed. Examples include GPS routing systems, the widespread toll collection systems around the world, and driverless cars under development by various known car manufacturers. Beyond those well-known applications, another field of "road automation" is also undergoing development, that is vehicular communication.

Vehicular networks enable a lot of applications. Besides the broad future of integrating Internet connectivity which provides entertainment and browsing activities, a core part of vehicular networks' functionality is to offer driving assistance. The drivers will benefit from vehicular communication enabled driving safety and driving efficiency enhancement. Examples of safety applications are collision warning, signal violation warning, and overtaking warning. Efficiency enhancement, on the other hand, is achieved by increasing traffic fluidity. Examples are traffic light optimal speed advisory, and co-operative navigation[20][44].

Vehicular communication is the wireless communication between vehicles, where there is no central router controlling the packet flow, thus is also called vehicular ad-hoc network(VANET). This kind of vehicular communication, however, sometimes requires assistance from existing techniques, like servers located somewhere on the Internet storing information for vehicles. Those servers need to have some access points sitting at the roadside to enable realtime queries of vehicles. The access points are called roadside unit (RSU) in vehicular communication. Thus the vehicular networks are generally considered to have two kinds of communication: including the vehicle-to-vehicle(V2V) com-

munication and vehicle-to-infrastructure(V2I) communication[1], together they are called V2X communication. A wireless communication technology to enable V2I communication that is often mentioned is Dedicated Short Range Communication(DSRC), which uses a frequency band in the 5.9 GHz range[20].

To facilitate vehicular communication, there are some hardware equipments to prepare. The vehicles will have on-board computation and memory resources (denoted as "on-board unit", OBU), and an antenna for wireless communication. It is also expected that there are some roadside units(RSUs) standing at the roadside working as access points and providing information for vehicles passing by.

Possible communication modes for V2V are versatile, including broadcast, unicast, geocast (a special kind of broadcast), and multicast[20]. Safety and efficiency applications, however, mainly use broadcast[45]. Both one-hop broadcast and multi-hop broadcast are used. The messages containing safety or efficiency enhancement information are broadcasted by the vehicle periodically and frequently to ensure they reach the largest number of relevant receivers in a region, usually within a range of a few hundreds of meters. The frequency of the repeated messages is around 1Hz to 10Hz[44].

The broadcasted messages often contain the current position of the vehicle sending the message if the broadcast is one-hop, e.g., the Cooperative Awareness Message (CAM)[45]. Beyond position, speed, heading, and other status information of the vehicle are all included in the message, which is sometimes denoted as a "heartbeat message" or "beacon" in literature. An example of such messages is shown in Figure 1.1, with the position in this message to be somewhere in Amsterdam. This characteristic of beacons originally fits the need of the nodes to know the location and current status of their neighbors. However, it also brings a concern that the vehicle can be tracked. Beyond the tracking problem, there are still some classical security problems for V2V communication if there is no security solutions. The messages may be tampered or forged, attackers may spread fake safety or efficiency enhancement warnings, private vehicles may pretend to be public-role vehicles, e.g., emergency vehicles, to gain privileges.

To prevent the possible security problems, security requirements need to be fulfilled. There were investigations in security requirements of V2X communication [43, 23]. To sum it up, the main security requirements are listed below:

1. Authenticity. Authentication of the legitimate participants in V2X communication is required. This means, on one hand, authentication of the vehicles should be ensured. On the other hand, if there are infrastructures participating in the communication, the infrastructures should be authenticated. Authentication of vehicles sometimes has more specific requirements other than simply authenticating the identity of the vehicle. For example, a public-role vehicle needs to prove its public role in order

[1]Some documents use the word I2V which means the communication is from infrastructure to vehicle

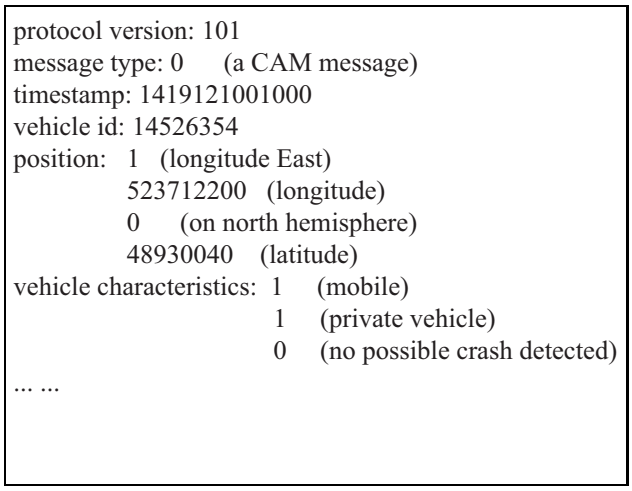

Figure 1.1: An Example Showing Part of A CAM Message

to gain some privileged use of roads. Or the vehicle needs to show that its claimed position is its actual position.

2. Integrity. Integrity of messages broadcasted by vehicles and RSUs should be protected. Also, the data stored in OBUs should not be able to be tampered with.

3. Confidentiality. Confidentiality is mainly required by unicast. And data stored in OBUs also needs to be protected from unauthorized access.

4. Availability. Functionality of vehicles and RSUs should not be held back if they are legitimate users. This is mainly required to prevent the denial-of-service(DOS) attacks.

5. Non-repudiation. This is required in case accidents or disputes may happen. For example, to find the reason of a crash, polices want to examine the messages sent before the time the crash happened. In this example, vehicles can not deny that they have sent a message if they actually have sent it.

6. Privacy. The location data comprised in the message can break the privacy of the driver and passengers in the vehicle. Because people may not want others to know their traveling locations. The privacy requirement, however, sometimes conflicts with other security requirements like authenticity and non-repudiation. This leads to the development of privacy-preserving authentication schemes which are introduced in Chapter 3.

In this thesis, we focus on the solutions that are devised to solve the privacy issue as well as to neutralize the contradiction of privacy and other se-

curity requirements. Another perspective is that the privacy requirements are constrained by basic system requirements like real-time constraints, robustness requirement, and scalability[36]. Those aspects also deserve attention when devising a privacy-preserving authentications scheme.

1.2 Privacy in Vehicular Networks

As mentioned in the previous section, privacy is a part of the security goals. However, one might argue that privacy is not so important and thus it can be ignored to reduce the costs. Here we give a brief summary about privacy and why privacy is important in vehicular communication.

1.2.1 Privacy in Digital World

What is privacy? Different cultures and contextual environments may have different definitions and goals of privacy. As mentioned by Westin in 1970,

> "Privacy is the claim of individuals, groups and institutions to determine for themselves, when, how and to what extent information about them is communicated to others"[42]

And in [29], privacy is defined as

> " Privacy is the right of an entity - in this context usually a natural person - to decide for itself when and on what terms its attributes should be revealed."

Privacy is so important that governments actually set it as a legal requirement. The EU data protection laws[2] and US Privacy Act are examples of that. Also, there are public organizations concerning people's privacy, like Privacy International[3] and World Privacy Forum [4].

The concern on people's privacy increased with the development of electronic and digital products, and since the development of networks. It is harder to protect people's privacy in the digital environment because people are likely to not even be aware of the privacy intrusion when that is happening on them, unlike in physical world[46]

With the growing use of networks in people's lives, privacy has drawn more attention from the public and academia. There are projects and solutions that aim to protect the privacy of people in computer networks. Examples are the European PRIME and PrimeLife projects[5] and Microsoft U-Prove technology[6].

[2]EU has launched series of data protection laws: Directive 95/46/EC, Directive 97/66/EC, Directive 2002/58/EC

[3]https://www.privacyinternational.org/

[4]http://www.worldprivacyforum.org/

[5]PRIME is the predecessor of PrimeLife. The websites of PRIME and PrimeLife are https://www.prime-project.eu/ and http://www.primelife.eu/

[6]http://connect.microsoft.com/site1188

As a part of private information, location data is also considered important for people. Researches have been conducted to fight against misuse of people's location data. For example, attacks on untraceability in Radio Frequency Identification(RFID) communication protocols have been analyzed[41][40].

1.2.2 Privacy Threat in Vehicular Communication

Due to the high mobility and frequent daily usage of vehicles, information about the movement of the vehicles has a high impact on privacy of people. As the location information of the vehicles can be used to deduce the movement of the driver and of the people in the vehicle, if there is no protection on the location information, the movement of people can be revealed. For example, a private vehicle that has been parked in the parking lot of a hospital can have high probability to result in the conclusion that the driver has been to the hospital. If a driver goes to the hospital frequently, it may be inferred that the driver or the one of the driver's family members is ill. If the previous example is not so appealing consider another example that when a vehicle "disappear" from the vehicular networks around one location at the time between 6-8 o'clock in the evening every workday, it can be inferred that the driver's home is just around that location because this time is for people to go home after work. And from the locations that the vehicle has been to on its way to go home, a path of the vehicle that it often follows can be concluded.

Although tracking attack mainly falls into passive attack, e.g., eavesdropping, it can also be used as a tool to collect information of the vehicle before the attacker may launch further attacks. Further attacks can be more offensive to the vehicle, e.g., impersonation, tampering and DOS attack. A real-world attacker may even intentionally cause traffic accidents toward a person more easily because he knows the vehicle of the person would show up at specific place and time.

With V2X applications, vehicles can be tracked easily if there is no effective solution to protect it. This is due to the ease of mounting an attack to trace vehicles. Firstly, vehicular communication is based on IEEE 802.11p communication protocol, which is a variant of the popular IEEE 802.11 wireless communication technology. Secondly, the tool to launch an attack is easy to find, such as a laptop or an access point from an evil or compromised service provider who could use this access point to do something else rather than providing the service. Thirdly, physical attack is also possible toward a specific OBU[17].

The attackers under discussion mainly fall into two categories: individuals who have limited computation and communication power, or governments and organizations that have large groups of computation and communication facilities, extensive monitoring scope even with the control of RSUs. It may be observed that by controlling public resources or powerful servers, individuals like terrorists can also launch the same level of attack as organizations. We contribute this kind of attack to the "governments and organizations" category.

In the first intuition, both kinds of attackers can track vehicles based on

Attacker Category	Computation Power	Threats
Individuals	Limited resource	Individual or small group tracking
Governments and organizations	Extensive resource	Individual and large-scale tracking, movement patterns and resolution profiling inference

Table 1.1: Privacy-Infringing Attackers in Vehicular Communication

the broadcast messages. The difference lies in the size of groups that can be tracked. Individual attackers are more likely to track individuals or small group of vehicles, whereas organizations can track large-scale group of vehicles, except from tracking individual and small groups. Another difference between individual attackers and organizations and governments is the motivation. Individual attackers may have their own target, like a celebrity or people they know of. And organizations and governments have no specific target at first, but they view all vehicles being monitored as possible targets, the tracking information may be stored in large databases waiting for real-time monitoring or future investigations. Beyond tracking, movement patterns of vehicles can be inferred if enough information on the tracked vehicles have been gathered. Finally, for large-scale tracking, high resolution profiling of individuals can be achieved if the tracking information can be linked with identities[36]. This is especially true for private vehicles. The different attacker models are listed in Table 1.1.

Now consider the scenarios like a private investigator following his target objects, a journalist following a celebrity, or an insurance company collecting statistic data of movement patterns of vehicles[15]. These kinds of privacy-infringing behaviors can be exacerbated without protection schemes.

Of course the tracking problem does not only harm privacy, it could also result in other problems. For example, criminals who track law enforcement vehicles to escape from being caught. The possible negative impact of tracking calls for solutions to prevent tracking.

Even if privacy protection mechanisms are in place, there could still be privacy infringing problems. In many privacy protection schemes, there is still an authority who has the ability to link messages to the identity who sends the message. This identity resolution ability is favored by law enforcement agencies when dispute happens in traffic accident. However, it is possible that this ability is misused. For example, the police can use this ability to search for vehicles who exceeds speed limit. Car manufacturers have a concern that this kind of scenarios can reduce the public acceptance of vehicular communication applications. An extreme privacy protection goal is to treat authorities as potential attackers and thus use cryptographic mechanisms to prevent authorities from breaching user privacy. This is called "privacy from a CA". Nevertheless, most if not all, privacy-protection broadcast authentication schemes do not consider privacy from CA.

1.3 Current Solutions

There are two intuitive approaches to prevent tracking. The first is to eliminate the usage of location data in broadcast messages, which is not feasible in V2X communication because many applications need the location data of the vehicle. The second is to hide the identity of the vehicles so that the location data can not be linked with the identity of the vehicle. The second approach is commonly used in vehicular broadcast authentication schemes.

To hide the identity of vehicles, a "bad behavior" that should be avoided is incorporating unique identifiers of the vehicles in broadcast messages. Not only a unique ID can reveal a vehicle's identity, when traditional digital signatures are used, a public key also resembles an identifiable token of a vehicle. This is extremely true when the popular authentication solution – Public Key Infrastructure (PKI) is chosen[12]. In PKI, there is a one-to-one mapping from the unique ID to the public key of a user. Thus traditional PKI does not satisfy privacy protection requirements.

Due to the deficiency of traditional authentication methods, a lot of new authentication schemes are brought out for vehicular communication. In Chapter 3 we describe two types of authentication schemes, namely pseudonym system(PS) and group signature(GS).

A pseudonym is a "an arbitrary identifier of an identifiable entity, by which a certain action can be linked to this specific entity", which is usually "a fictitious name" of the entity[29]. PS protects the user's privacy in a way that message receivers do not see the identity of the message originator, but only see the pseudonyms of the originator. A pseudonym in PS is often a public key that can be used to verify a signature which is attached to a message. Pseudonyms are preloaded by vehicles, and usually are issued by pseudonym issuers. Pseudonyms of a vehicle are changed to prevent long-term tracking which happens when a pseudonym is used for a long period. It is not extensively discussed how often a pseudonym should be changed, however some benchmarks use the cycle of pseudonym changes as high as 3 to 60 seconds[7].

Obviously the messages sent by the same pseudonym are linkable, making the vehicle trackable in the lifetime of a pseudonym. Another issue is frequent changing of pseudonyms may incur much overhead on pseudonym issuer and on vehicles. Moreover, in many PS schemes, the pseudonym issuer knows the pseudonyms it issued to vehicles, that means the pseudonym issuer is able to track vehicles.

GS, on the other hand, can be viewed as a method to achieve anonymity. In [29], anonymity is defined as "the quality or state of being not identifiable within the set of all possible entities that could cause an action and that might be addressed". In vehicular communication, anonymity implies each two messages of the same originator are unlinkable. GS is used for a group of vehicles, e.g., vehicles in a district, who have different private keys only known by themselves and who have a common public key for all vehicles in the group. Thus vehicles can sign messages with their own private keys and verify signatures using the common public key. In this way it is not feasible to link messages which are

signed by the same private key.

Although PS and GS are useful in hiding identities of vehicles as well as in authentication, they generally do not realize attribute authentication, and usually the attributes(e.g., the age of the driver, the type of the vehicle, and the size of the vehicle, etc.) of the vehicles are not discussed at all. This means certain services, like toll collection and fleet management, would require other solutions rather than reusing the broadcast authentication scheme (these services are not vehicular communication, though). Also, classification of public-role and private vehicles calls for modification of existing schemes.

Now the question is do we have an authentication scheme to classify vehicles in vehicular communication, and at meantime the authentication scheme can be reused in other services, while vehicles can avoid being tracked only because they reveal their position information in broadcast messages. We seek the solution from attribute authentication, which we introduce in the next section.

1.4 Attribute Authentication

Here we define the methods of authentication into several categories according to the functionality of the methods.

1. Entity authentication: There are different definitions of entity authentication. In [28], Entity authentication is defined as "the process whereby one party is assured (through acquisition of corroborative evidence) of the identity of a second party involved in a protocol, and that the second has actually participated (i.e., is active at, or immediately prior to, the time the evidence is acquired)". In [29], entity authentication is defined as "the corroboration of the claimed identity of an entity and a set of its observed attributes". In conclusion, entity authentication is the authentication of the identity of the other party. The method of entity authentication varies between the widely used password verification, PKI based certificate verification, challenge-response authentication, and biometric recognition. Obviously a unique ID is mandatory in this category of authentication, otherwise the other party is not identifiable.

2. Pseudonym authentication: Pseudonym authentication is a variant of entity authentication, in that pseudonyms(fictitious names or random numbers) are used in the authentication scheme, rather than a real ID of the other party. The advantage of pseudonym authentication lies in its hiding of the identity of the other party, which is required for privacy protection reason. Many PS schemes realizing pseudonym authentication have been brought out in the past decade(see Chapter "Related Work").

3. Message authentication: Message authentication is a means to make sure that the message is from the claimed originator, and that the integrity of the message has not been tampered during transmission[28]. Message authentication is also mentioned as data authentication, which is defined as "the corroboration that the origin and integrity of data is as claimed" in [29]. Message authentication codes are widely used for message authentication. When the message receiver turns to be all nearby nodes of the message sender in a network, we

call this kind of message authentication as broadcast authentication. Broadcast authentication is the authentication of the originator of a broadcasted message. In vehicular communication, we mainly focus on the authentication of broadcast messages.

Traditionally, message authentication requires the originator to reveal its unique ID, since otherwise there is no way to link the message to the originator. However, for privacy protection reasons, a unique ID is unfavorable. Moreover, people want the messages sent by same originator to be indistinguishable from messages sent by other users (see unlinkability in chapter 2). Thus traditional message authentication schemes do not apply for privacy protection reason.

4. Attribute authentication: Contrary to entity authentication, attribute authentication does not necessarily need the identity of the participator. Attribute authentication does not use pseudonyms either. Instead, the attribute or combination of several attributes of the other party is examined. An example of attribute authentication is a vehicle belonging to a certain type, e.g., truck, car, ambulance, etc. Or the size of the vehicle falls into a certain interval. Attribute authentication is investigated in PRIME and PRIMELIFE project[1], where anonymous credentials are used to realize attribute authentication.

By using attribute authentication instead of entity authentication, attribute authentication achieves anonymity. This characteristic makes it a candidate to meet the goal of tracking avoidance and privacy protection in V2X broadcast authentication. Another characteristic that makes it superior is attribute authentication carries more information than other message authentication schemes, since attributes of the identity are also included in the authentication. One might argue that by injecting attributes like birthday, name, health status in a traditional PKI certificate, entity authentication can also carry a lot of personal information. And as a variant of entity authentication, pseudonym authentication can carry as much information as entity authentication. However, attribute authentication is more flexible in the sense that it is user-controlled. Users can choose to reveal one part of personal information while hiding the other. Whereas injecting attributes in traditional PKI certificates or pseudonym certificates should reveal all information about the entity at a time. For example, by using attribute authentication, a person reveals that he was born in a region of Netherlands, say, Twente, without revealing that he was born in 1979 or any other information.

Because attribute authentication carries more information than merely showing that one is a legitimate user, and because it is flexible and user-controlled, attribute authentication achieves a kind of integrated authentication. That is, one certificate serves many applications. For example, a vehicle reveals that it is a Toyota car with a specified generator while connecting with a server offering remote diagnostic service, whereas it shows it belongs to a fleet while entering a parking lot for that fleet, all by using the same certificate issued to this car. For broadcast authentication, it shows that it is a legitimate private vehicle without revealing its identifier or other information, or it shows that it is a public-role vehicle in order to gain high privilege of the road.

Although anonymity that attribute authentication brings is a powerful solu-

tion to avoid tracking, it also has some side effects on V2X applications. Because of anonymity, some applications like data aggregation do not work well, because a vehicle can claim to be more than one entities to gain higher trust in a majority voting based scheme(See "Sybil Attack Suppression" in Chapter 2.

There are only a few attribute authentication schemes, mostly devised for Internet transactions. CL-Idemix is one of the more efficient and mature attribute authentication schemes. CL-Idemix is introduced in Prime and PrimeLIFE project[4]. CL-Idemix employs CL signature[10] to achieve attribute authentication. However, CL-Idemix assumes Internet environment and thus is not directly usable in vehicular communication. The main problems is that CL-Idemix is an entity authentication scheme. It does not consider message authentication, nor can it be used in broadcast authentication. Instead, CL-Idemix requires to set up a session between two nodes. It is our work to tune CL-Idemix to suit the need of message broadcast of vehicular communication.

1.5 Thesis Structure

The thesis is structured as follows: In Chapter 2, we listed the requirements for a secure and privacy-preserving broadcast authentication scheme. We carefully select and divide the requirements into basic and optional ones to separate the core requirements as well as to enable extensions. In Chapter 3, we narrate and discuss the existing broadcast authentication schemes in vehicular communication. We categorize those schemes and evaluate them toward the requirements. In Chapter 4, we introduce the preliminaries and show how CL-Idemix works in Internet environment. In Chapter 5, the CL-Idemix based Broadcast Authentication scheme(CLIBA) which is used in vehicular communication is described. In Chapter 6, we show our implementation and performance of CLIBA and analyze the security of our scheme. Finally in Chapter 7, we summarize the result of the thesis and decide on future work.

Chapter 2

Requirements of Broadcast Authentication in V2X

2.1 List of Requirements

There should be criteria to analyze existing authentication approaches with respect to their suitability in VANET, and to devise new authentication approaches. The criteria can be set by the requirements for an authentication approach to be privacy-preserving broadcast authentication scheme. We come up with a list of requirements which are based on and collected from existing research results, including papers of security requirements in VANET[23][36] and various authentication schemes that have been brought up (illustrated in Chapter 3). We divide the requirements into the basic ones and the optional ones, in which the basic ones are the necessary conditions for a scheme to be secure and privacy-preserving. The optional ones are the extension from the basic ones, and hence are not necessary conditions.

The basic requirements are:

1. Message Authentication Without Originator Verification. In IVC, there is a huge demand on message authentication, since the safety message broadcast is driving the need for a secure and efficient authentication scheme to verify the messages. However, traditional message authentication does not meet the privacy protection goal. Thus a new kind of message authentication that does not reveal the identity of the originator is what we want.

2. Attribute Authentication. The authentication scheme can realize attribute authentication, i.e. allow to attest certain car attributes, like the car being an emergency vehicle and being allowed by some authority to participate in IVC.

3. Privacy Protection. Do not leak any privacy infringing information about the sender of messages, such as a unique ID.

4. Strong Unlinkability. Being able to link two or more messages together to decide if they come from the same originator should be avoided in the highly mobile inter-vehicular communication settings, since then the location privacy of the originator is violated (heartbeat messages contain location data of the originator). We discuss the degree of unlinkability in section 2.2.

5. One-hop Broadcast Authentication. Either do not allow any broadcast back channel, which indicates that the transmission of the authentication message is one-way and with no intermediate nodes, or allow only a broadcast back channel in other vehicles' broadcast messages, which means the transmission of authentication message can be back and forth with no intermediate nodes (but is limited to broadcast). This means there is no interactive protocols but messages should be self-contained so that the recipient can perform authentication itself.

6. Small Size. The authentication information should be lightweight to not overload the communication medium. According to [11], the V2V packet should be less than 100 bytes. So the size of the authentication information is supposed be no more than 100 bytes to make it applicable in practice. In that case we need to consider asymmetric crypto mechanisms with a small signature and certificate, or circumvent asymmetric cryptography by clever use of symmetric cryptography. However a problem of the 100-byte standard is for many broadcast authentication scheme 100 bytes are not enough(see Chapter 3). Nevertheless, it is better to always bear in mind that a scheme with smaller package size is more favorable than a scheme with larger package size in broadcast authentication.

7. Low Computation Overhead. The delay of the authentication procedure should be small. For the life critical applications, the delay is even more precious. Whereas the smallest "maximum latency time" of applications scenarios defined in the ETSI document[44] is 50 ms, this latency time includes the time of processing and communication of a message from the sender to the receiver. So the time allocated for the authentication steps is even smaller. Moreover, since there are usually more verification than signature generation processes for a vehicle, the signing time of the scheme could be longer than the verification time.

8. Independent Authentication. Do not require a permanent connection with any TTP or other infrastructure component. Intermittent communication with TTP might be possible in a configurable interval. The interval is supposed to be no less than one day, better interval lengths might even be months or years. In the ideal case, no such communication is necessary at all.

The optional requirements are:

1. Resolution of anonymity. Resolution of anonymity is the disclosure of the identity of the originator of certain messages usually generated by

misbehaving or malfunctioning vehicles when traffic accidents or disputes happen. While literature extensively mix resolution and isolation (see optional requirement 2) together into the concept of "anonymity revocation", dividing the two notions can elaborate the resolution process. The reason why we set them as optional requirements is due to the legal background, some countries do not have a clear legal attitude toward resolution of anonymity, such as the EU countries, whereas other countries support such a resolution, for example the US. If resolution of anonymity is included, this mechanism should be protected from abuse by various attackers, including authorities.

2. Isolation of Vehicle. After resolution of anonymity, isolation is conducted by authorities to exclude the specified vehicle from the system, so that the legitimate vehicles do not trust the vehicle anymore. The time interval between the isolation of vehicle starts and the isolation completes should be small to exclude the vehicle as fast as possible. This time interval, named isolation time interval, is introduced in [18].

3. Non-repudiation. Non-repudiation is required in message authentication, aiming that the originator of the message should not be able to deny having sent the message. Non-repudiation is based on the assumption that the resolution of anonymity is feasible. Otherwise there is no target identity, i.e., no originator, for the non-repudiation property.

4. Sybil Attack Suppression. Sybil Attacks are prevented, i.e., prevent a vehicle from massively replicating its presence in the network. Sybil attacks are used by an attacker to win in a majority voting based data aggregation scheme and security mechanisms. In some pseudonym credential based authentication schemes, an attacker may use multiple pseudonyms to launch Sybil attacks. If this optional requirement is needed, then such a pseudonym credential based authentication scheme is not qualified.

5. Multi-hop Authentication. The broadcast message can be relayed by neighbors to receivers out of the broadcast range of the originator. In that way the message is completely uni-directional and there should be no back channel at all. Obviously multi-hop authentication fulfills one-hop authentication automatically.

6. Context Based Authentication Attribute usage could be limited to context (time, position, orders, etc.). As an example, imagine a police car that is only allowed to use a "right-of-way" attribute while on duty.

The reason to divide the requirements into two parts is to separate the mandatory requirements from the optional ones. In that way the authentication schemes that also fulfill the optional requirements are more advanced than the authentication schemes that only fulfill the basic requirements. And the authentication schemes that do not fulfill all basic requirements do not qualify

as secure and privacy-preserving broadcast authentication scheme as defined in this paper.

In literature there are some recommendations for security and privacy requirements in V2X communication. In [23], a series of security requirements for VANET security are collected. Since the paper is not dedicated to the privacy problem, the requirements are not fine-grained in privacy.

In [36], a set of fine-grained and layered requirements are brought up, concerning privacy and its dependencies on system and other security aspects, and the inter-relations among the requirements are analyzed. Many of the requirements are also used or similar in this paper. For example, the authentication requirement is similar with the basic requirement 1 in this paper, anonymity is similar with the basic requirement 3 in this paper. There are also differences between the requirements. The unlinkability requirement in [36] is quite different from basic requirement 3 in this paper. And the real-time constraint requirement has a more accurate definition in this paper, as is shown in basic requirement 6 and 7.

2.2 Unlinkability Degree Determination

In C2X communication, there are two intuitive criteria to decide the unlinkability degree, namely linkable time and linkable number of messages. Linkable time is the length of time during which the messages sent by the same originator can be linked with the probability to be 1. Linkable number of messages are the number of messages sent by the same originator linkable by any receiver with a probability of 1. Linkable number of messages and linkable time are transferable if we know the number of messages sent in a time unit. For example, if we change a vehicle's id after 1000 messages, and if the vehicle send 10 messages every second, then the linkable time of the vehicle is 100 second. Generally in this paper we use linkable time as a measurement.

In [7], a linkable time of 3 to 60 seconds is used in simulation of a VANET, showing a satisfactory result when several optimizations are made to the original authentication scheme. The shorter the linkable time, the stronger the unlinkability degree. The perfect linkable time is 0, that is, any two messages of the same originator are always unlinkable.

Being unlinkable does not mean that linking is impossible. On the contrary, linking is still possible, only with a degraded probability. Generally speaking, the linking probability depends on the size of anonymity set. Anonymity set is the "set of all possible subjects who might have sent a message"[3]. Linking probability with respect to the anonymity set is a topic in the "mix zone" research, which does not related much with our purpose. We introduce shortly the "mix zone" in Chapter 3.

Chapter 3

Related Work

3.1 State of the Art

There are many protocols and cryptographic systems proposed for privacy-preserving authentication in IVC. For each proposal, the terminology may be different from each other. To analyze the proposals, we will unify the terminologies used in all the proposals. We will use "scheme" to refer the proposed protocols and systems. The content to be signed in a message is called payload. The term "verifier" and "signer" are also named "receiver" and "originator" in different context.

3.1.1 Schemes

In this subsection, we describe the main features of the schemes under investigation.

The most intuitive approach to realize privacy-protection broadcast authentication is used in **SeVeCom**[32][24]. In this scheme, vehicles receive pseudonyms and the credentials of the pseudonyms from trusted authorities in a secure channel. The pseudonyms are public keys for the vehicles to use in broadcast authentication, and credentials are just signatures on the pseudonyms by the trusted authorities. Accompanying the public keys are the corresponding secret keys for the vehicles to sign messages, which are held secret by the vehicles. The vehicles can use one pseudonym for a period of time, which is under control of a hardware security module (HSM).

It can be seen that **SeVeCom** realizes pseudonym authentication. However, the way it realizes pseudonym authentication is like a traditional PKI infrastructure. The only difference between **SeVeCom** and PKI is that **SeVeCom** issues pseudonym credentials and PKI issues identity certificates. Trusted authorities work as pseudonym providers(PPs), which are required to verify the long-term identity of a vehicle before issuing pseudonyms. The PPs are placed at roadside or can be connected through Internet.

Based on **SeVeCom**, **V-tokens**[35] further enhances privacy protection by separating the roles of certificate authorities(CAs), PPs, and resolution authorities(RAs). Note that in **Sevecom** the tasks of all the three roles are performed by PPs, so there is no RA or CA in it. CAs issue credentials of v-tokens for vehicles. V-tokens are randomized ciphertexts which hide the identities of the vehicles and which can reveal the identities of the vehicles only by the RAs. More specifically, a v-token is an encrypted message using the public key of RA, in which the message contains the vehicle id, the id of CA who issues this v-token, and a random number. A vehicle uses a credential of v-token to request a pseudonym from a PP. Then the PP checks the credential, extracts and leaves the v-token in the issued pseudonym. The broadcast authentication process is more or less the same with **SeVeCom**, while the identity resolution process incorporates more than one RAs to engage in a secret-sharing homomorphic decryption scheme (like ElGamal[16]).

Another scheme, which claims to be an upgrade of PKI, namely **PKI+**[47], is adopted in vehicular communication system in [2]. For privacy protection concern, [2] suggests using pseudonyms issued under **PKI+** in all layers of communication.

PKI+ does not distribute pseudonyms for the vehicles. Instead, vehicles generate their own pseudonyms from their master keys, which are chosen by themselves and certified by the certificate authority. **PKI+** utilizes advanced cryptography, such as bilinear paring and zero-knowledge, to realize pseudonym and message authentication without originator verification. Since **PKI+** asks vehicles to issue their own pseudonyms, there is no PPs in this system.

ECPP[27] is also a pseudonym based system, which uses the PPs to generate pseudonyms and pseudonym credentials for the vehicles. Like in **SeVeCom** the long-term identity is also verified by PPs before issuing the pseudonyms. The difference with **SeVeCom** lies in the methodology it uses, **ECPP** is more complicated because it utilizes advanced cryptographic methods(see Section 3.1.2 to know the difference).

Sun's IDB[39] and **Kamat's IDB**[22] utilizes identity-based(IDB) cryptography to realize pseudonym authentication. In the two schemes, the vehicles request PPs to generate IDB secret and public key pairs that they would use in the broadcast authentication process in a period of time. In IDB cryptography, the public key is also the identifier of the owner of the key. The originator uses an IDB secret key to sign a message, and attaches the public key as a pseudonym after the signature. Then the verifier uses the public key to verify the signature.

SRAAC[18] is a pseudonym scheme which involves multiple servers to issue pseudonyms to vehicles. Hence the resolution of anonymity also requires multiple servers.

Unlike previous schemes,**GSIS**[26] is not a pseudonym scheme. But it realizes message authentication without originator verification by group signature. In **GSIS**, a vehicle registers at the membership manager to acquire its private key in a group(e.g., a territorial region), with which the vehicle signs messages. The verifiers verify the signatures of the originator using public information, without knowing any specific information about the originator.

In [7] three types of pseudonym schemes are described and performance of those schemes are measured. In this paper we analyze the **Hybrid** scheme and do not consider the other two schemes, because the other two schemes are basically similar with the aforementioned schemes. The **Hybrid** scheme utilizes group signature to let the vehicles sign their own pseudonyms. Verifiers can verify the pseudonyms via the group signature scheme, and then use the pseudonym to verify the message signature. In this way the vehicle can choose their own pseudonyms and decide for how long their pseudonyms are alive. However, the **Hybrid** scheme can incur heavy overhead. To reduce the overhead, the authors have used several optimization methods. The main idea of the various optimizations is to use group signature only once or only for the first several messages to let the receivers receive and verify the pseudonym which is signed via the group signature scheme. The remaining messages do not involve group signature, but only requires the verifier to verify the message signature using the pseudonym that it has received in the first or the first several messages.

There are other broadcast authentication schemes brought out, which are based on symmetric cryptography. However, those schemes use a unique ID, thus do not provide privacy protection. For example, TESLA[34] and its offsprings [38]. We do not consider those schemes in this paper.

3.1.2 Methodology

The schemes under investigation to implement privacy-preserving authentication have a common feature, that is, they all use message signatures. The difference is how the signature is generated. Thus the schemes can be categorized according to the methods used. There are two main categories, namely pseudonym system (PS) and group signature (GS). The common feature of PS schemes is that a temporal public key is used as a pseudonym of the vehicle. In that way the temporal public key has two roles: a temporal id of the vehicle, and a public key for signature verification. The way how such a temporal public key is generated leads to a two-level categorization. The full methodology graph is in Figure 3.1.

The schemes normally use a combination of various methods to fulfill their expectation on the system goals. For example, secret sharing is used to share a key among multiple authorities in resolution of anonymity (**Sun's IDB**, **SRAAC**, **V-tokens**). The reason for key sharing lies in the goal to prevent abuse of the resolution ability.

Another method **SRAAC** and **V-tokens** use except secret sharing is blind signature. In **SRAAC**, blind signature is used to ask the PPs to issue pseudonyms unknown and unpredictable by the PPs themselves. Whereas in **V-tokens**, blind signature is used to blind the v-token (and other validity information of v-token) so that the v-token is unknown by the CA when the CA issues a credential of the v-token. Blind signature protects the privacy of the vehicle to the extent that the pseudonym or identifiable information is protected even from the authorities, in that way only the vehicle itself knows its pseudonym or any identifiable information before using it.

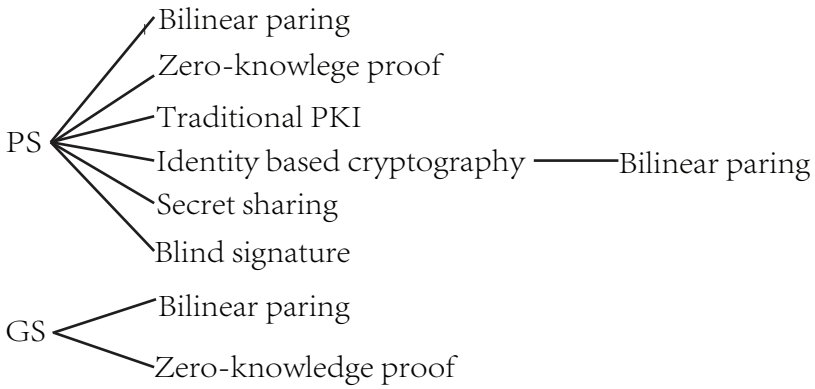

Figure 3.1: The methods category

Identity-based cryptography is another good way to realize PS (**Sun's IDB**, **Kamat's IDB**), since it eliminates credentials of pseudonyms, which result in much smaller size of authentication information.

As we mentioned in Chapter 1, the common way to prevent tracking in vehicular communication is to hide the identities of vehicles. Since in message authentication, public keys are deemed as the elements to link the vehicles' identity, the goal of hiding the identity of a vehicle can be achieved in two ways, that is, to randomize the temporal public key of a vehicle so that any two messages signed by different temporal public keys of a same vehicle can not be linked to that vehicle(PS schemes), or to use a generic public key for all vehicles so that all message signatures can be verified using the same public key, but any two messages signed by the same private key can not be linked(GS schemes).

For the first way of hiding identities, the main building blocks for randomization of temporal public key include zero-knowledge proof and blind signature, while bilinear paring acts as auxiliary tool to reach for authentication goal. For the second way of hiding identities, it mainly utilizes group signature. The schemes under investigation and the cryptographic primitives they use are listed in Table 3.1.

It should be pointed out that, although the **Hybrid** scheme utilizes group signature, it is not a GS scheme. Because the scheme embeds group signature in traditional PKI, the main architecture resembles traditional PKI.

3.1.3 Evaluation of the Schemes

At the first glance, we are supposed to evaluate the schemes individually according to their fulfillment of the requirements set in the previous section. But when they fall into the two main categories of PS and GS, they have common properties in fulfillment of the requirements. We have summarized those common properties of the PS and GS schemes as follows:

For the basic requirements,

Scheme	Category	Crypto Primitives
PKI+[47]	PS	bilinear paring, zero knowledge
ECPP[27]	PS	bilinear paring, zero knowledge
Hybrid[7]	PS	traditional PKI, group signature
SeVeCom[32]	PS	traditional PKI
V-tokens[35]	PS	blind signature, secret sharing
Sun's IDB[39]	PS	identity-based signcryption
Kamat's IDB[22]	PS	identity-based signcryption
SRAAC[18]	PS	secret sharing, blind signature
GSIS[26]	GS	group signature

Table 3.1: Crypto Primitives of the Schemes under Investigation

1. Message Authentication Without Originator Verification. All schemes implement message authentication. PS schemes reveal the temporary identity, that is the pseudonym of the message originator. But since the pseudonyms are changed after a short period, long-term identity of the originator is not revealed. GS schemes does not reveal the originator of any message.

2. Attribute Authentication. None of the schemes implement attribute authentication. But when we think of traditional PKI as a way to embed attributes in certificates, then **SeVeCom** and **Hybrid** can be seen as attribute authentication schemes.

3. Privacy Protection. All schemes provide privacy protection. The only difference is how and to what extent they provide privacy protection, which is measured by unlinkability level.

4. Strong Unlinkability. For PS schemes, the pseudonym lifetime is adjustable. Strong unlinkability can be achieved by choosing a short pseudonym lifetime, since then the linkable time of the messages sent by the vehicle is short. Some PS schemes include a timestamp in the pseudonyms(**ECPP**, **V-tokens**, **Kamat's IDB**). The pseudonym by timestamp mechanism has two derivations. One is the timestamp, which indicates the valid period of the pseudonym, is previously set by the pseudonym provider(PP)(**ECPP**, **V-tokens**). The other is let the verifier decide on a trusted lifetime threshold (**Kamat's IDB**).

   For GS schemes, the unlinkability level is extremely high, since the message signature changes due to the random elements injected in signature creation. The unlinkability level of GS schemes equals to using one pseudonym per message, i.e., linkable time of 0. In that case, the linking probability depends on the size of the anonymity set. For group signature schemes, the size of the anonymity set is the size of the group.

5. One-hop Broadcast Authentication. All schemes support one-hop broadcast authentication. The schemes can also be extended to multi-hop broadcast applications easily since their authentication procedures are unidirectional and thus do not require a back channel.

6. Small Size. See section 3.1.4 for discussion of message size.

7. Low Computation Overhead. we discuss the overhead in section 3.1.4.

8. Independent Authentication. All schemes support independent authentication.

For the optional requirements,

1. Resolution of anonymity. All schemes support resolution of anonymity. Due to consideration of abuse of the resolution ability, some schemes provides resolution by collaboration of multiple authorities through secret sharing (**SRAAC**, **Sun's IDB**). The scheme of **ECPP**, however, implements resolution of anonymity through the collaboration of the trusted authority who has an identity database, namely an identity manager, and the RSU who has issued the pseudonym credentials. **V-tokens** propose both of the two ways mentioned above, the only difference is it involves collaboration of the RAs and the CAs in the second way, rather than the identity manager and RSU.

2. Isolation of Vehicle. All schemes fulfill isolation of vehicles. There are two kinds of isolation solutions, namely pre-issuing and post-issuing isolation. Pre-issuing isolation aims to stop the issuing of a new pseudonym, for PS schemes. Post-issuing isolation, however, aims to stop the verification of the message signature or pseudonym credentials already issued, for both PS and GS schemes. The isolation behavior of the various schemes is summarized in Table 3.2 (**Sun's IDB** does not have a clear description of its isolation method).

   The intuitive approach to do isolation for PS schemes relies on distribution of certificate revocation list (CRL), or revocation list (RL) if there is no pseudonym credential in the scheme. CRL or RL contains the identifiable information (such as a unique ID) of the revoked vehicle or simply revoked pseudonyms. When CRL or RL is distributed among PPs, the PPs would use the RL to decide on pseudonym requests from vehicles, which implements pre-issuing. When the CRL or RL is distributed among vehicles, the vehicles check received pseudonym against the RL.

   Pre-issuing and post-issuing isolation with CRL or RL have both advantages and disadvantages. On one hand, post-issuing isolation with CRL or RL incur delay and memory overhead for the vehicles, caused by the distribution of RL to the vehicles. On the other hand it also benefits from a shorter isolation time interval. Indeed, compared with pre-issuing isolation, post-issuing isolation can distribute the updated RL before the

Scheme	Isolation of Vehicle
PKI+	pre-issuing(updated information), post-issuing(RL)
ECPP	pre-issuing(RL)
Hybrid	post-issuing(RL)
SeVeCom	pre-issuing(CRL), post-issuing (RL)
V-tokens	pre-issuing(CRL)
Sun's IDB	–
Kamat's IDB	pre-issuing(RL)
SRAAC	pre-issuing(RL)
GSIS	post-issuing(RL, self-updating)

Table 3.2: Isolation Behavior of the Schemes

revoked pseudonyms expire. But it also depends on how fast the updated RL can be distributed to the vehicles. A generic scheme for CRL distribution shows at most 30-40 minutes delay for receiving a updated CRL[33]. In that case, it is useless in pseudonym systems with pseudonym lifetime to be less than 30 minutes. The RL can be distributed in a centralized way, i.e., by RSUs and Internet servers, or can be distributed in a distributed way, i.e., among vehicles. **SeVeCom** and **PKI+** suggest both ways, whereas **PKI+** has a much smaller size of RL (linear to the number of revoked vehicles).

Pre-issuing isolation with CRL or RL does not lay burden on vehicles, but it can make the isolation time interval long, e.g., a misbehaving vehicle continues sending verifiable messages until it uses up all pseudonyms.

There are other ways of isolation except through CRL or RL. **PKI+** enables pre-issuing isolation through the PP updating its public and private keys and publishing the updated public key. The normal vehicles are supposed to update their own keys according to the published public key of PP, whereas the revoked vehicles can not get valid new keys because the new public key of PP has excluded them from the system(i.e., isolation of the vehicles). The crucial point is the vehicles need updated keys to create valid pseudonyms for themselves.

Like **PKI+**, **GSIS** also provides isolation through both RL and updated information. The difference lies in how to combine the two ways. While **PKI+** has the two ways run in parallel, **GSIS** apply the two ways under different conditions. When the number of revoked vehicles is smaller than a threshold, a revocation verification algorithm is used by the verifier to check the validity of the signer against a RL. When the number of revoked vehicle is larger than the threshold, the vehicles all need to update their private keys according to the RL, while the revoked vehicles can not update. The size of the RL is $171 * R$ bits, R is the number of revoked vehicles, which is applicable in practice.

3. Non-repudiation. All schemes support non-repudiation. However, for **V-tokens**, non-repudiation is incomplete. Because blind signature is used when authentication of a vehicle toward a CA is conducted in the v-token credential issuance process, there is no 100% confidence to ensure that the vehicle and its v-tokens are authentic.

4. Sybil Attack Suppression. All schemes except **SeVeCom** are vulnerable to Sybil attack. Most schemes do not consider hence do not prevent Sybil attack. **SeVeCom** uses a hardware security module(HSM) to forbid using more than one pseudonym at a time. For those PS schemes who allow vehicles to compute pseudonyms by themselves(**PKI+**), there is no restriction on how many pseudonyms can be computed at a time. GS schemes also does not prevent Sybil attack due to their extreme unlinkability. Other PS schemes do not have restrictions on how many pseudonyms a vehicle can request from a PP or from different PPs, neither do they have restrictions on how many pseudonyms a vehicle can use at a time.

3.1.4 Overhead Comparison

Authors of the schemes commonly use different methods to evaluate their schemes, which makes it difficult to compare the overhead of different schemes. Hence a unified method should be utilized to compare those schemes towards the same standard.

Authentication Information Size

The public key and signature size in different schemes have different evaluation methods. In this paper we compare the authentication information size of all schemes according to the order (using the symbol p of the cyclic group or field, in which the processes of various schemes take place. In some schemes, there are two signing schemes in use. We denote the orders the two cyclic groups as p and q respectively. We also eliminate the information such as lifetime of pseudonym, id of PP, since these data do not have a consolidated size.

The size of the authentication information is calculated based on the following criteria:
1) Because many broadcast authentication schemes in vehicular communication use ECDSA as the signing method, we apply ECDSA for the schemes which do have a signing procedure and which have not specified signing methods. For those that have specified their own signing methods, we use the specified signing methods.
2) As an illustration of common use, for ECDSA based scheme, since ECDSA with a security level of 80 bits is commonly used in many schemes, which results in group order of 160 bites (20B), we compute the authentication information size of the ECDSA based schemes when the group order is 20B. For the non-ECDSA based schemes, we compute the authentication information size according to a comparable group order with security level to be 80 bits.

Scheme	Signing Scheme	Size	when security level is 80 bits
PKI+	discrete logarithm problem	8p	160B
ECPP	ECDSA	9p	180B
Hybrid	group signature: in [5] , message signature : ECDSA	$7p + 3q$	209B
Hybrid -with optimization	message signature : ECDSA	$2p$	40B
SeVeCom	ECDSA	5p	100B
V-tokens	message signature: ECDSA, v-token: Elgamal	5p+2q	140B
Sun's IDB	in [21]	2p	40B
Kamat's IDB	ECDSA	2p	40B
SRAAC	ECDSA	5p	100B
GSIS	discrete logarithm problem	9p	180B

Table 3.3: Size of Authentication Information

The expected sizes of authentication information for the schemes is listed in Table 3.3. There is a sharp contrast among the schemes. The IDB schemes of **Sun's IDB** and **Kamat's IDB** have smallest sizes, whereas the group signature based schemes of **Hybrid** and **GSIS** have the biggest size. Note that when the **Hybrid** scheme is optimized under various optimizations, it has a shorter size of authentication information to be 40B. The IDB schemes benefit from the cutting off of the pseudonym credentials.

Computation Overhead

In C2X communication, signature verification has a more strict demand on the computation time than signature generation, since vehicles are supposed to verify a lot more messages than what they generate. Hence the verification speed is more important than the generation speed. Another problem in comparing the computation overhead of the schemes is that the papers usually use different machines and cryptographic library to implement their schemes. Moreover, the security level(key length) they choose are different.

In some papers, the authors add the computation time of all kinds of costly operations(e.g., point multiplication, bilinear paring)used in their schemes, and count the sum of the time. The estimated time of a kind of operation comes from a source of third party. We apply this method to compare the computation overhead of all schemes. In the cryptographic computation, the most costly operation is point multiplication, bilinear paring, multiplicative inverse, and exponentiation. We list the number of those operations in Table 3.4. We use pm to represent point multiplication for elliptic curve, E for exponentiation

Scheme	Signing	Verification
PKI+	$1E$	$8E + 1P + 2Inv$
ECPP	$1pm$	$11pm + 3P + 1Inv$
Hybrid	$1pm + 1Inv$	$6E + 5Inv + 2pm + 3P + (1Inv + 2P) * N$
SeVeCom	$1pm + 1Inv$	$4pm + 2Inv + c * N$
V-tokens	$1pm + 1Inv$	$4pm + 2Inv$
Sun's IDB	$1E$	$1E + 1P$
Kamat's IDB	$2pm$	$1pm + 2P$
SRAAC	$1pm + 1Inv$	$4pm + 2Inv$
GSIS	$6E + 1P$	$8E + 2P + (3P + 1Inv) * N$

Explanation of acronyms: E– exponetiation. pm– point multiplication for elliptic curve. Inv– multiplicative inverse. P– bilinear paring. N– number of entries in a RL. c– time to check one entry in the RL.

Table 3.4: Computation Overhead of the Schemes

Scheme	Environment	Crypto Scheme	Sig. Time	Verif. Time
ECPP	Intel Pentium IV 3.0GHz	MNT curve $k = 6$, 160 bit q	–	$21.88ms$
Kamat's ID-B	667MHz G4 PowerPC	curve $y^2 = x^3 + x$ in 512-bit finite field	$116.6ms$	$124.2ms$
Hybrid	1.5GHz Centrino	GS with security level 128 bit, ECDSA security level 96 bit	$54.2ms$	$52.3ms$
Hybrid-with optimization	1.5GHz Centrino	ECDSA security level 96 bit	$0.5ms$	$3ms$ (do not check RL)

Table 3.5: Computation Overhead Data Collected from the Papers

(multi-exponentiation is broken into exponentiations), Inv for multiplicative inverse, P for bilinear paring, N for the number of entries in the RL, c for the time to check one pseudonym in the CRL.

We list the estimated computation time of the schemes in Table 3.5, which are collected from the original papers.

If we disregard multiplicative inverse,and exponentiation, and if we use the 3.0 GHz machine which is utilized in [27], the point multiplication costs 0.6 ms, and bilinear paring costs 4.5 ms on MNT curve(k=6, 160 bit q). The schemes using pure elliptic curve have the estimated computation time as listed in Table 3.6. The time for signing and verification meet the time constraint of IVC broadcast communication. However it is obvious that the verification time is much longer than the signing time, which is opposite to the requirement.

Scheme	Signing	Verification
ECPP	$0.6ms$	$21.88ms$
SeVeCom	$0.6ms$	$2.4ms + c * N$
V-tokens	$0.6ms$	$2.4ms$
Kamat's IDB	$1.2ms$	$9.6ms$
SRAAC	$0.6ms$	$2.4ms$

Explanation of acronyms: N– number of entries in a CRL. c– time to check one entry in the CRL.

Table 3.6: Estimated Computation Overhead of the Schemes Using MNT Curves

3.1.5 Mix Zone

Even if the privacy preserving PS schemes and GS schemes help protect the identity of the vehicles cryptographically, the driving behavior of the vehicles and the geographical conditions can reveal some "side-channel" information to let the attacker predict the next possible location after pseudonym changes. For example, a vehicle driving on a straight road can be predicted with high probability its next location after a very short period. The direction of a vehicle passing a cross road can be computed with high probability from the speed of the vehicle and the different lengths of the routes to turn around or go straight ahead. To analyze such "side-channel" information, a notion of mixzone was brought up.

The basic idea of a mixzone is a spatial area where no location-aware applications are available and where the mobile entities change their pseudonyms [3]. The SeveCom project uses the technique of mixzone to solve the problem of tracking of vehicles. When the notion of mixzone is applied in the vehicular network(VN), it is extended to fit in the environment of VN as follows: In [6], a physical area, e.g., a city, is divided into observed zones and unobserved zones, with observed zones being the district monitored the adversary, and all the unobserved zones together form a logical mix zone. Assume that the vehicles do not know where the observed zones are (or if they are in an observed zone), and that pseudonym change frequency is so high that the vehicle would surely change its pseudonym in an observed zone, and thus observed by the adversary. When the vehicles change their pseudonyms in an unobserved zone, that zone functions as a mix zone, where the adversary can not link the pseudonym of the vehicle entering the zone and the pseudonym of the vehicle leaving the zone. [3] and [6] both modeled effectiveness of mixzones mathematically and simulated the scenarios in their respective systems, but there was no actual method or protocol to implement the realisitic mixzone architecture.

A realistic protocol to implement the mixzone notion is, however, introduced in [19], in which a Cryptographic Mix Zone(CMIX) protocol uses RSUs to distribute symmetric communication keys for a "mixzone". The mixzone here is a spatial area within the broadcast distance of a RSU, in which the RSU validates

Schemes	Req.1	Req.2	Req.3	Unlinkability	Req.5	Size	Time (verification)	Req.8
PKI+	Y	N	Y	Flexible	Y	Fair	Fair	Y
ECPP	Y	N	Y	Flexible	Y	Big	Big	Y
Hybrid	Y	Y	Y	Flexible	Y	Big	Big	Y
SeVeCom	Y	Y	Y	Flexible	Y	Fair	Small	Y
V-tokens	Y	N	Y	Flexible	Y	Big	Small	Y
Sun's IDB	Y	N	Y	Flexible	Y	Small	Small	Y
Kamat's IDB	Y	N	Y	Flexible	Y	Small	Fair	Y
SRAAC	Y	N	Y	Flexible	Y	Fair	Small	Y
GSIS	Y	N	Y	High	Y	Big	Big	Y

Table 3.7: The Fulfillment of Basic Requirements

PKI based certificates of vehicles, and then distribute symmetric communication keys for vehicles to use in the mixzone. The vehicles broadcast messages encrypted with the same symmetric communication key distributed by the R-SU. Intuitively in this way the adversary can not link the identities of vehicles since the vehicles all use the identical symmetric communication key.

3.2 Summary

In the previous section, we evaluated the schemes against the requirements of a secure and privacy-preserving broadcast authentication scheme set in section 2.2. It can be concluded that except no scheme supports attribute authentication nor do they prevent Sybil attack, those schemes have fulfilled the other requirements, although the size of the authentication information or the computation overhead are high for some schemes, or the isolation methods may incur burden on the receiver side if post-issuing isolation is used. Context-based authentication is not discussed since it relies on attribute authentication. The summary of the result of the evaluation is shown in Table 3.7 and Table 3.8.

We also gave a short summary of the mixzone technique, which deals with a "side-channel attack" on location information of vehicles.

3.3 Open Problems

The already existing authentication schemes for C2X communication are suitable for broadcast authentication except they do not consider Sybil attack, and except the size of the authentication information is sometimes too large. However, they generally do not realize attribute authentication. As mentioned in Chapter 1, attribute authentication represents anonymity, which provides stronger privacy than PS. And since attribute authentication carries more information than merely telling that it is a vehicle, it can be used in more applica-

Schemes	Req.1	Req.2	Req.3	Req.4	Req.5	Req.6
PKI+	Y	Y	Y	N	Y	–
ECPP	Y	Y	Y	N	Y	–
Hybrid	Y	Y	Y	N	Y	–
SeVeCom	Y	Y	Y	Y	Y	–
V-tokens	Y	Y	N	N	Y	–
Sun's IDB	Y	Y	Y	N	Y	–
Kamat's IDB	Y	Y	Y	N	Y	–
SRAAC	Y	Y	Y	N	Y	–
GSIS	Y	Y	Y	N	Y	–

Table 3.8: The Fulfillment of Optional Requirements

tions than both PS and GS schemes. Our research question is to devise a secure and efficient attribute authentication scheme for V2X broadcast communication, which has an acceptable computation and communication overhead. We build our work upon existing approaches. As PRIME/PRIMELIFE[1] projects have investigated attribute authentication, a protocol suite named Idemix was brought up and extended through those projects. We would investigate this scheme and see if it can be utilized or adapted in the V2X environment.

We have a series of hypothesis as listed below:

1. The authentication protocol Idemix in PRIME/PRIMELIFE projects can be applied to V2X communication, or can be adapted for V2X communication. Thus a broadcast authentication scheme can be build upon the PRIME/PRIMELIFE Idemix protocol.

2. Such a scheme can satisfy the requirements as defined in the previous chapter("Requirements") at least as good as the schemes analyzed in this paper.

3. The scheme can prevent Sybil attack.

4. The scheme can be efficient in computation and communication overhead.

Those hypotheses will be investigated and answers to those hypotheses would be provided in the remainder of the thesis.

Chapter 4

CL Signature and Idemix

In this chapter, we first introduce the basic concepts and building blocks of the CL signature, then we describe the CL signature scheme and the Idemix protocol suite. The CL signature scheme is a signing scheme that enables a server to generate a signature blindly on a list of messages for a requested user. In this process some of the messages are provided by the user and are hidden from the server[10]. The CL signature scheme is named after its authors Camenisch and Lysyanskaya. The Idemix protocol suite is based on CL signature and extends CL signature to attribute authentication. Our purpose is to adapt the CL-Idemix attribute authentication in V2X communication, which is then described in Chapter 5.

4.1 Preliminaries

In this section, we explain some basic concepts that are required to understand CL signature and Idemix protocols.

Safe Prime. Safe primes p and q are primes that have the form of $p = 2p'+1$, $q = 2q' + 1$, with p' and q' are also primes.

Special RSA Modulus. A RSA modulus $n = pq$ with p and q are both safe primes is a special RSA modulus. It can be seen that the size of the RSA group is $\phi(n) = 4p'q'$. If we consider a subgroup of quadratic residues modulo n: $QR_n \subseteq Z_n^*$, the size of the subgroup is $|QR_n| = \frac{1}{4}\phi(n) = p'q'$.

Schnorr's Identification Scheme. Schnorr's identification scheme[37] is a simple three-way zero-knowledge proof scheme which proves the knowledge of a discrete logarithm x of a specific number y (mod n):

$$PK\{(m) : y = g^x\}$$

If we denote the prover as P, and verifier as V, the proof is as follows:
Step 1. P comes up with a random number r, computes $t = g^r$ mod n, and then sends t to V
Step 2. V comes up with a random challenge c and sends it to V.

Step 3. P computes $s = r + cx$ mod n, and send s to V. V verifies if $t == g^s y^{-c}$. If $t == g^s y^{-c}$, then V is convinced that P knows the discrete logarithm of y.

The above scheme is correct as

$$t = g^s y^{(-c)} = g^{r+cx}(g^x)^{-c} = g^r$$

The security of Schnorr's identification scheme relies on the hardness of discrete logarithm problem, i.e., the function $t = g^r$ is one-way.

Schnorr's identification can be turned non-interactive by Fiat-Shamir heuristic. The non-interactive scheme reduces the number of rounds of information exchange between P and V, and thus saves time and bandwidth in network environment. The non-interactive Schnorr's identification is achieved through making the challenge c a hash value of t:

Step 1. P comes up with a random number r, computes $t = g^r$ mod n,$c = h(t)$, $s = r + cx$, and send c, s to V

Step 2. V reconstructs t by applying $t = g^s y^{-c}$, and sees if $c == h(t)$. If $c == h(t)$, then V is convinced that P knows the discrete logarithm of y.

The correctness of the non-interactive scheme is shown as

$$h(t) = h(g^s y^{(-c)}) = h(g^{(r+cx)}(g^x)^{(-c)}) = h(g^r) = c$$

The function $h()$ is not required to be a one-way function here since $t = g^r$ is already an one-way function. The non-interactive version of Schnorr's identification scheme is used in Idemix and many other privacy-preserving authentication schemes, e.g., **PKI+**[47], **ECPP**[27] and **GSIS**[26].

4.2 CL Signature

CL signature is a signing scheme which enables a server to sign information provided by users[10].

Recall from previous section that a special RSA modulus n has the form of $n = pq$, with p and q both are safe primes. Take p and q as private keys, and randomly select three group members of the quadratic residue subgroup QR_n, namely a, b, and c. Publish a, b, c, n as public keys. If we want to sign a message m, we choose a random number w, and a prime e, then compute

$$A = (a^m b^w c)^{e^{-1} \bmod |\mathrm{QR_n}|} \mod \mathrm{n}$$

Since we know p and q, and p and q are safe primes, then we know p' and q'. As $|QR_n| = p'q'$, we know $|QR_n|$, which is used in the computation of A. Now we have the signature of m: $\{A, e, w\}$.

When we want to verify the signature, we see if

$$A^e == a^m b^w c \mod \mathrm{n}.$$

If the result is correct, then the signature is valid. Otherwise the signature is not valid.

For signing a block of messages $m_1, m_2, ..., m_L$, we choose as many a as m, and do

$$A = (a_1^{m_1} a_2^{m_2} ... a_L^{m_L} b^w c)^{e^{-1} \bmod |\mathrm{QR_n}|} \mod \mathrm{n}$$

The verification is obviously through checking

$$A^e == a_1^{m_1} a_2^{m_2} ... a_L^{m_L} b^w c \mod \mathrm{n}$$

The unforgeability of CL signature relies on the strong RSA assumption, which is shown in the following:

Strong RSA Assumption. Given RSA modulus n and an element $u \in \mathbb{Z}_n^*$, it is hard to compute values A and $e > 1$ such that $A^e = u \bmod \mathrm{n}$.

Although the original CL signature also uses zero-knowledge proof to prove that a user has a CL signature on a message without simply revealing the signature. Those functions are not used in Idemix. Instead, Idemix combines the original CL signature scheme with non-interactive Schnorr's Identification to do that. So we are not going to introduce the proving holdership of a CL signature process of original CL signature scheme in this thesis.

4.3 Idemix

Idemix [4] is a protocol suite which extends CL signature from a signing scheme to attribute authentication(also called "private credential system" in literatures). A java project which implements Idemix is developed and maintained by IBM Zurich lab[1].

Idemix considers the messages to be signed in CL signature as attributes of a certificate which is issued to the users by an issuer. Users can do attribute authentication with each other. As the value of a certificate is influenced by some random data, it changes every time when the certificate is re-issued even when the certificate contains the same attribute values. Thus a certificate in the system is also called a credential to distinguish it from traditional PKI based certificates.

A credential is a CL signature (A, e, v) which is signed by the issuer. The relation of the three elements A, e and v is slightly different from the original CL signature. It has the form of

$$A = ((a_1^{m_1} a_2^{m_2} ... a_L^{m_L} b^v)^{-1} c)^{e^{-1} \bmod |\mathrm{QR_n}|} \mod \mathrm{n},$$

with $\{m_1, m2, ...m_L\}$ as the attribute values.

The Idemix protocol suite has three parties: Issuer, Prover, and Verifier. Issuer initializes and manages the CL signature generation process to issue credentials for users. Users can prove that they have certain credentials signed by the Issuer, in which case the users are Provers. Consequently users can also verify that certain credentials are signed by Issuer, in which case the users are Verifiers. By proving, it means Prover does not show all the attribute values

[1] http://www.zurich.ibm.com/security/idemix/

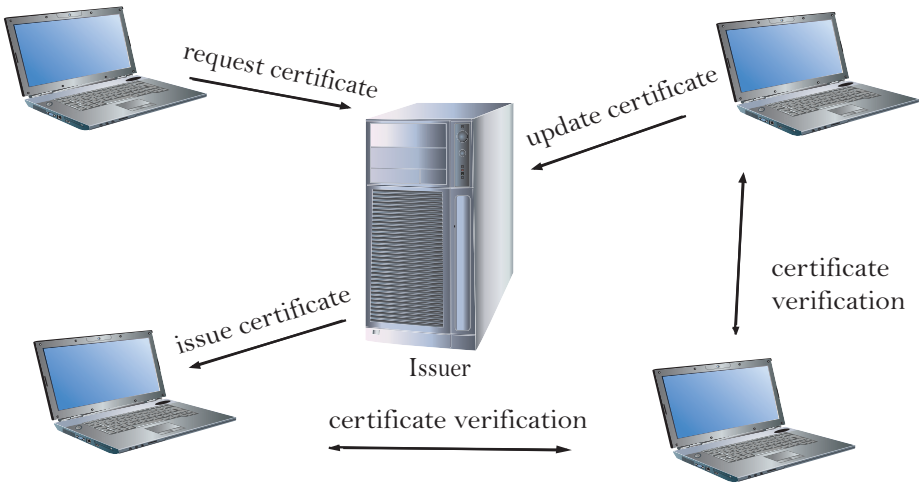

Figure 4.1: Idemix System Structure

that have been signed in the credential, but only proves that it has some valid credentials containing attribute values willingly shown by the prover, whereas values of other attributes are not revealed.

The system works as is illustrated in Figure 4.1.

The Idemix protocol suite has incorporated a series of protocols to realize attribute authentication. The whole process of attribute authentication is divided into several protocols, as shown in Figure 4.2. It should be mention that not all protocols are necessary. For example, during verification, only the protocols {build proof, ProveCL} and {verify proof, VerifyCL} are necessary, whereas the rest protocols are optional. Furthermore, most of the protocols in the system are too complex to be explained clearly in a short space in the thesis. Thus we only describe the main procedure of the main protocols of Idemix, which consists of the protocols {build proof, ProveCL} and {verify proof, VerifyCL}. For a thorough understanding the readers should read the reference paper of Idemix.

1. **System Setup**. This is the first step of Idemix, and is not deemed as a protocol in later version of specification document of Idemix [48] since there is no interaction in this step of the specification document. In this step, Issuer initializes the system. Specifically, Issuer creates and publishes attribute specification, and sets up the CL signature environment by choosing private and public keys and various system parameters. The attribute specification specifies the attribute characteristics and the order that the attribute values are included as in the credential element A. The system setup protocol can be

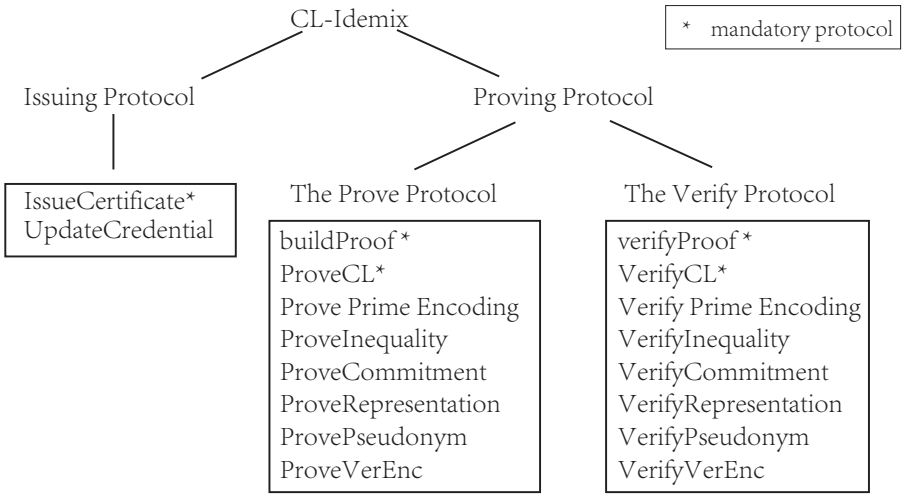

Figure 4.2: Idemix Protocol Suite

moved to vehicular communication directly, because it only involves Issuer.

2. **Certificate Issuance and Update**. Issuer issues credentials for users. The issuance starts when a user requires a credential from the Issuer. The user engages with Issuer in an interactive protocol to let Issuer sign a credential for the user using CL signature.

Attributes in a credential are divided into three types, namely **user controlled attributes** A_{UC}, committed attributes A_{CO} and **known attributes** A_k. Idemix utilizes non-interactive Schnorr's Identification that makes A_{UC} and A_{CO} hidden from the issuer, and thus they can be grouped as **hidden attributes** A_h. Issuer knows A_k values of the user, and it does not know A_h values. The values of A_{UC} are chosen and controlled by the users. The name of A_{UC} is used to differentiate itself from A_k. As A_k values are known by Issuer, it can also be seen as they are chosen and set by Issuer. A_{CO} attributes are used for commitment, in which a user commits a value to be signed by Issuer and waits to open the commitment (by revealing the value that is committed) in future time. The commitment scheme used in Idemix is introduced in [13]. Since commitment is not expected to be used in vehicular broadcast, we omit the commitment scheme in this thesis.

Here we give an example of A_{UC} and A_h in vehicular communication. A vehicle has a list of attributes $Attr = \{key, vid, role, type, alias\}$, of which key is a secret key of the vehicle, vid is the vehicle id, $role$ is the vehicle being private or public vehicle, $type$ is the vehicle being a car, a bus or a truck, etc.

And *alias* is an alias of the vehicle, which is used as an identifier of vehicle that the driver of the vehicle has registered for some third-party services. The value of *key* and *alias* are controlled by the vehicle. So

$$\{key, alias\} \subset A_{UC}.$$

The vehicle tells Issuer the value of *vid*, so Issuer can find values of *role* and *type* for that vehicle. In this case *vid*, *role* and *type* are known attributes to Issuer:

$$\{vid, role, type\} \subset A_k.$$

It is assumed that the user and Issuer have already done an authentication process and have securely exchanged necessary data, including the A_k values. And then the user combines values of A_h in the form of $\prod a_i^{m_i}$. Note that this is a compound form of $y = g^x$ in Schnorr's Identification. Later on the values of A_{UC} and A_{CO} are proved by the user in non-interactive Schnorr's Identification scheme. The combined form of A_h and the proof data are sent by the user to Issuer. Issuer would verify the non-interactive Schnorr's Identification proof data sent by the user. If the proof of the user is valid, Issuer stores the values of all three types of attributes in a credential (A, e, v) which has the form mentioned before. Because the values of A_{UC} and A_{CO} only exist in C and commitments, Issuer does not know those values.

A similar scenario with issuance is credential update. When some attribute values changed, a user might want to update its credentials, which involves the credential update protocol. The credential update protocol is basically a part of the issuance protocol.

3. **Credential Verification**. In this protocol, Prover proves that it has a valid credential, and shows the attribute values that he would like to reveal, or proves that the attribute values meet certain conditions. The conditions could be the value being in a range, or that the attribute value is the same or different from the attribute value of another valid credential. Consequently Verifier verifies all the claims made by Prover. We introduce the main process of the verification protocol here to give the reader a glance on how the CL signed credentials and attribute values are proved and verified. To simplify the description, we do not consider the scenario where Prover proves the attribute values meet certain conditions. The main procedure of attribute authentication extends Schnorr's Identification, and is illustrated in Figure 4.3.

Before Prover starts the proof, he first sets up a connection with Verifier and exchanges the proof specification and the session number n_1. The proof specification is a list of claims that Prover would prove. In the proof specification, a set of attributes that Prover would reveal is defined. The set is denoted by **revealed attributes** A_r. On the other hand, the other attributes would not be revealed, and they are grouped in **unrevealed attributes** $A_{\bar{r}}$. Then values of A_r are sent by Prover.

Here we give an example of A_r and $A_{\bar{r}}$ following the example in the Issuance part again. For a list of attributes $Attr = \{key, vid, role, type, alias\}$ in a credential, normally the vehicle does not need to reveal values of all attributes, so

all attributes belong to $A_{\bar{r}}$ by default.

$$\{key, vid, role, type, alias\} \subset A_{\bar{r}}$$

When a police car broadcasts its presence and requires all cars nearby to reveal their *vid*, then *vid* is a revealed attribute for those cars.

$$\{vid\} \subset A_r, \quad \{key, role, type, alias\} \subset A_{\bar{r}}$$

When a public vehicle is performing a public task, it reveals this attribute (which has a value of "public"). So *role* by default is an unrevealed attribute, and only when public vehicle sets its *role* attribute to be a revealed attribute, it gains privilege of the road.

$$\{role\} \subset A_r, \quad \{key, vid, type, alias\} \subset A_{\bar{r}}$$

The $type, alias$ value can also be revealed when there is a third-party service, like parking.

$$\{type, alias\} \subset A_r, \quad \{key, vid, role\} \subset A_{\bar{r}}$$

Now Prover wants to prove that he has a credential (A, e, v), and values of revealed attributes A_r are contained in the credential. The proving is basically non-interactive Schnorr's Identification. Prover first computes **Common** which is a set of common values that are needed in the later process, **T** which is a list of t-values, and *FuncData* which are data for functionalities like commitments on values of some unrevealed attributes. Here a t-value serves as the value t (for unrevealed attributes) in non-interactive Schnorr's Identification as is introduced in Section 4.1. The commitment scheme used here is the same as in Issuance protocol. We omit the proving of committed attribute values here again for simplicity.

For each unrevealed attribute $m_i \in A_{\bar{r}}$, Prover computes an s-value:

$$s_{m_i} = r_{m_i} + cm_i,$$

in which c is of the form

$$c = H(context, FuncData, \mathbf{Common}, \mathbf{T}, n_1).$$

Here *context* is a list of public parameters. Prover stores all s-values in a set **s** and sends the list of $\{c, \mathbf{s}, \mathbf{Common}\}$ to Verifier.

For Verifier to verify that Prover has a credential (A, e, v), and values of A_r are contained in the credential, he first reconstructs **T** and *FuncData*. The reconstruction of **T** is basically the second step of the non-interactive Schnorr's Identification. Then c is calculated based on reconstructed **T** and *FuncData*, plus the received **Common**, and n_1 which was previously generated by Verifier itself. Verifier compares it against the hash value c he received. If they are equal, then Verifier is convinced that Prover has proved its claims.

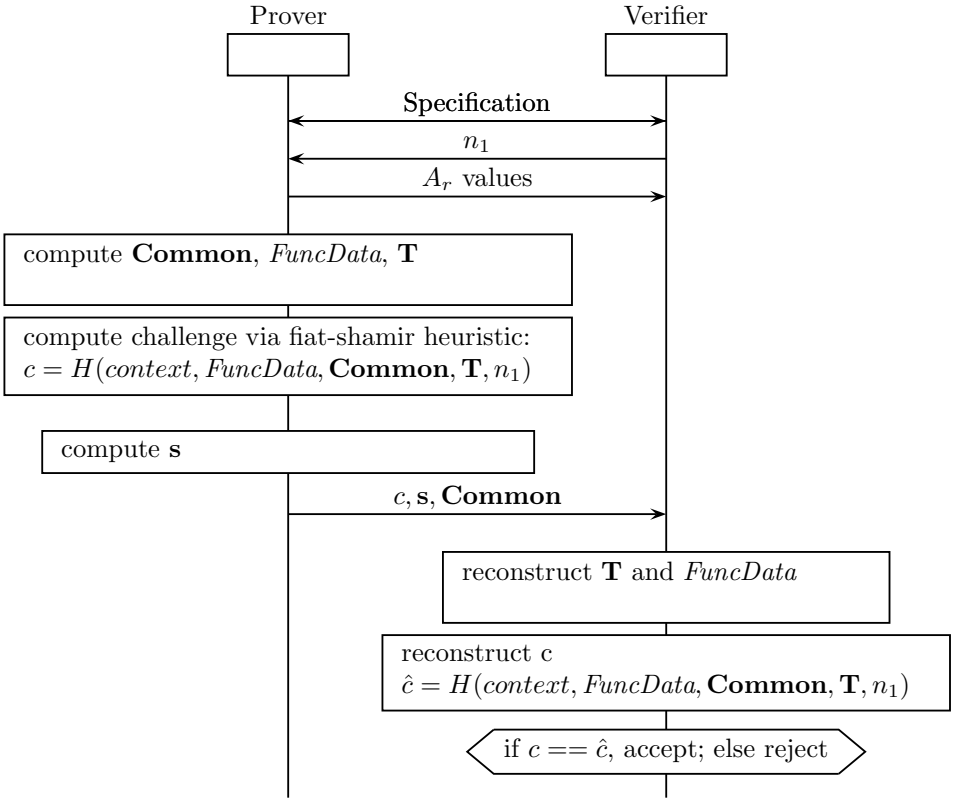

Figure 4.3: Idemix : Proof of valid certificates

Chapter 5

The CL-Idemix Based Broadcast Authentication Scheme - CLIBA

To use CL-Idemix in VANET environment, we need to answer questions like "is it applicable to use CL-Idemix in VANET" and "how can we tune it to fit in VANET better". In this chapter, we first evaluate the CL-Idemix scheme according to the requirements we set in Chapter 2. Then we make modification and enhancement to CL-Idemix to suit the need of VANET environment. Finally we describe the system that we build in a macroscopic view.

5.1 Using CL-Idemix in VANET

The building protocols in Idemix can in principle be re-used in or be transferred to vehicular communication environment.

Firstly, for the issuance and update protocol, Issuer can be a remote server that vehicles can connect with through wifi access points, RSUs, or cellular network. It is noted that the vehicle and Issuer need mutual authentication before Issuing process and a secure connection is established. The issuance protocol is performed under the secure connection. The result of the issuance is that the known attribute values and the credential are sent to the vehicle from Issuer. Possible attributes of vehicles include the vehicle type (car, bus, or van), the vehicle role (public or private), the valid time of the credential, and the signing key of the vehicle which is used to sign the messages (c.f. Section 5.2.1). After the new credential is acquired, vehicles can update their certificates through the update protocol if the known attribute values are changed. For example, when the valid time of the certificate is about to expire, the vehicle run the update protocol to extend the certificate. Those scenarios are included in Figure 5.1. In the figure, the upper part shows the scenario where the vehicles

are running on a road. Those vehicles exchange messages with the RSUs, the cellular access point(cellular AP), and with each other. With RSUs and cellular AP, vehicles update their credentials. The RSUs and cellular AP work as the agencies, and are further connected with the Issuer server. With the other vehicle, the vehicle verifies credential of the other vehicle. The lower part of the figure shows the scenario where vehicles park near some buildings, e.g., houses and hotels. The vehicles connect with the WiFi network of the building. And the Wifi network is connected with Internet, in which the Issuer server resides. In this way, vehicles can request or update a credential from the Issuer server through the WiFi network of the building.

Secondly, for the credential verification protocol, vehicles can do attribute authentication with each other in a vehicle-to-vehicle(V2V) fashion without TTP, which corresponds to basic requirement 2 and 8 as we mentioned in Section 2.1("Attribute Authentication" and "Independent Authentication" correspondingly). The most common usage of the credentials in V2V is to prove that the entity who broadcasts the messages is a vehicle with a valid credential signed by the Issuer. As an special example, imagine that an ambulance or a fire engine broadcasts its presence to gain privilege over the road, or a police car sending a command to request all vehicles around to give their ids (maybe in encrypted form).

Thirdly, because the credential verification protocol fulfills attribute authentication with a zero knowledge property, basic requirement 3 ("Privacy Protection") is also reached. And the unlinkability degree is high. As each time the prover's credential is proven rather than shown, random elements are injected to compute the proof value, the linkable time for each credential proof is 0.

However, for the verification protocol, although Idemix uses non-interactive Schnorr's Identification scheme for zero-knowledge proof, it is not fully non-interactive. Because it is required that Verifier sends a session number to Prover before Prover starts the zero-knowledge proof to ensure the freshness of the connection and the freshness of the proof created by Prover. If this does not completely breach basic requirement 5 ("one-hop broadcast authentication") in Section 2.1, it at least places a burden on network.

Another issue that prevents credential verification of Idemix from being applicable in vehicular communication lies in the fact that Idemix does not fulfill message without originator authentication, which is basic requirement 1 in Section 2.1. Because Idemix is an identity management tool rather than a message authentication scheme. Furthermore, it is assumed to work in Internet, where mature message authentication schemes already exist, such as SSL and TLS [14].

Moreover, the authentication information size of the verification process of Idemix is quite large, and the computation overhead is still big (see analysis in Section 5.1.1).

The full lists of evaluation of Idemix according to the basic and optional requirements are shown in Table 5.1 and Table 5.2, respectively. It is noted that even if most of the basic requirements are fulfilled, most of the optional requirements are not.

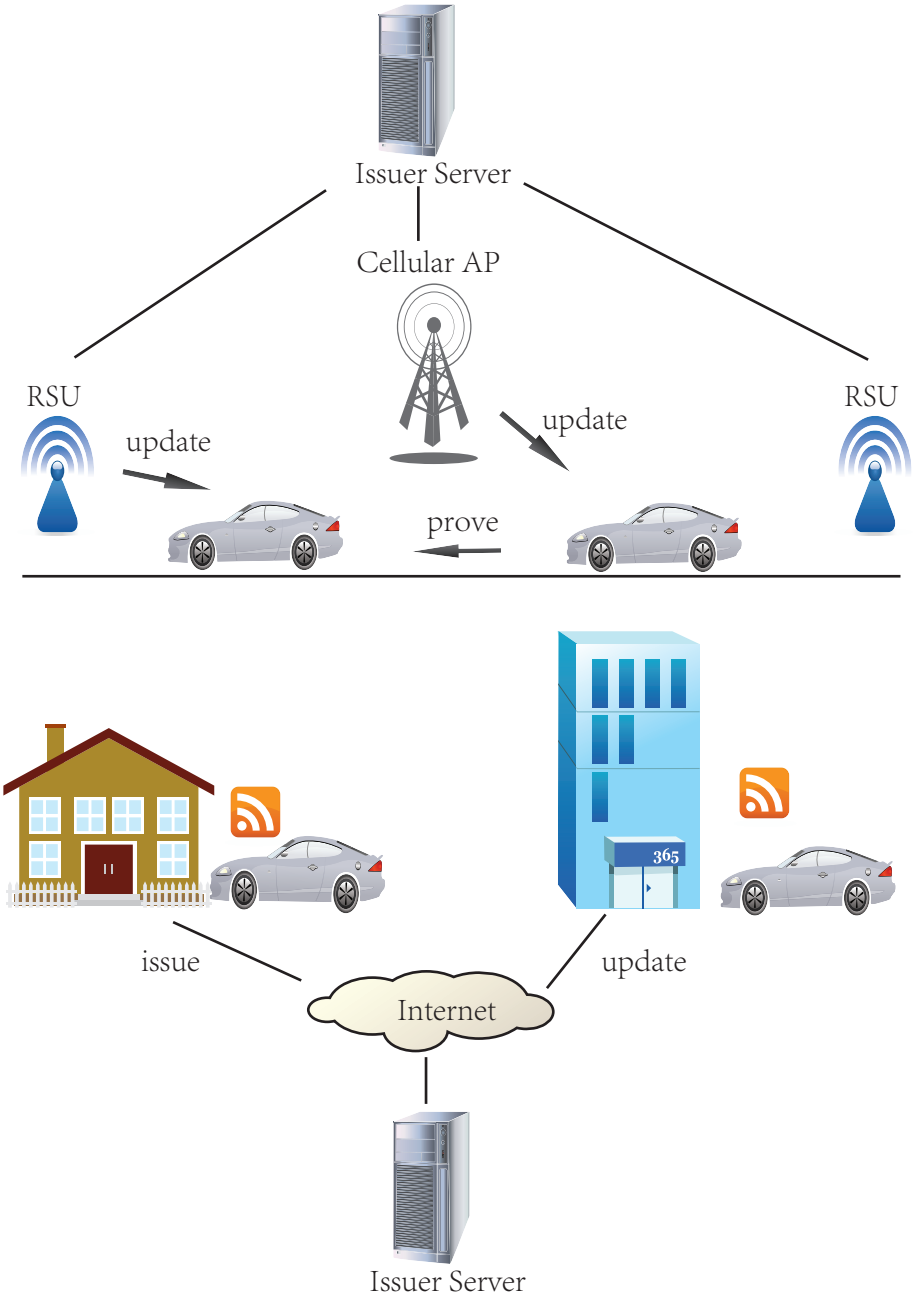

Figure 5.1: Using Idemix in VANET

Scheme	Req.1	Req.2	Req.3	Unlinkability	Req.5	Size	Time (verification)	Req.8
Cl-Idemix	N	Y	Y	High	N	Large	Big	Y

Table 5.1: The Fulfillment of Basic Requirements by CL-Idemix

Schemes	Req.1	Req.2	Req.3	Req.4	Req.5	Req.6
Cl-Idemix	N	N	N	N	Y	N

Table 5.2: The Fulfillment of Optional Requirements by CL-Idemix

5.1.1 Idemix Overhead

Authentication Information Size

The authentication information comprises of the the data transferred during execution of the verification protocol, e.g., n_1 , A_r values, c, **s** and **Common**. Here we give an estimation of the size of the proof of a user owning a certificate, not including the proving of committed attribute values and proving of attribute values meeting certain conditions. The s-value for proving that a user owning a certificate has a bitlength of

$$l'_e + l_v + (2 + N_{m_i \in A_{\bar{r}}})l_\phi + (2 + N_{m_i \in A_{\bar{r}}})l_H + N_{m_i \in A_{\bar{r}}} l_m + N_{m_i \in A_{\bar{r}}} + 4.$$

The meanings of the parameters are listed in Table 5.3 below. And the bitlength values for those parameters that are used in Idemix Java implementation are also shown in the table. The size of the s-value for a credential reaches $5567 + 593N_{m_i \in A_{\bar{r}}}$ bits, which is $691.1 + 74.1N_{m_i \in A_{\bar{r}}}$ bytes.

Besides s-value, c has a bitlength of l_H, **Common** has a bitlength of l_n, A_r values have a bitlength of $N_{m_i \in A_r} l_m$. So the total size of data that Verifier would receive (excluding n_1 since n_1 is sent by Verifier) is

$$l_n + l'_e + l_v + (2 + N_{m_i \in A_{\bar{r}}})l_\phi + (3 + N_{m_i \in A_{\bar{r}}})l_H + N_{m_i \in A} l_m + N_{m_i \in A_{\bar{r}}} + 4,$$

which results in $947.1 + 74.1N_{m_i \in A_{\bar{r}}} + 32N_{m_i \in A}$ bytes in the Java implementation project of Idemix. Using the example in Section 4.3, when Prover has $A_{\hat{r}}$ and A_r as

$$\{type, alias\} \subset A_r, \quad \{key, vid, role\} \subset A_{\bar{r}}$$

The proof is 1330 bytes. Even if this is only the size of the proof that a user has a certificate, it can be seen that the authentication information size of Idemix is quite large, compared with the authentication schemes reviewed in Section 3.1.4, where the biggest authentication information size is around 200 bytes, and the smallest authentication information size is only 40 bytes. The large size of the authentication information may cause the packets to be so big that there are more collisions in high density network. It can be foreseen that when the

Parameter	Meaning	Bitlength used in Idemix
l'_e	size of interval that e values are taken from	120
l_v	size of the v values of a credential	2723
l_ϕ	security parameter that governs the statistical zero-knowledge property	80
l_H	domain of the hash function used in Fiat-Shamir heuristic	256
l_m	size of attributes	256
$N_{m_i \in A_{\bar{r}}}$	number of unrevealed attributes	–
l_n	size of RSA modulus	2048
$N_{m_i \in A_r}$	number of revealed attributes	–
$N_{m_i \in A}$	number of attributes in the certificate	–

Table 5.3: Meanings of Parameters in Size of Proof

size of revealed attributes and the proof of other functionality, e.g., the attribute value falling in a range are included, the credential proof is even larger.

Computation Overhead

If we only consider the proving and verification of a user owning a certificate, not including the committed attributes and the proving of attributes meeting certain conditions, the computation overhead of proving is

$$(1 + 3N_{cert})E,$$

and the verification overhead is

$$3N_{cert}E.$$

Here E means the time for modular exponentiation, N_{cert} means the number of certificates (in previous discussion we assume N_{cert} to be 1), and we also break multi-exponentiation into a number of exponentiations, just as we did in Section 3.1.4. However, since the RSA group size is large, the computation overhead is not deemed to be low compared with other schemes reviewed in Section 3.1.4.

5.2 Enhancing CL-Idemix in VANET

CL-Idemix fulfills attribute authentication, which has not been proposed for use in VANET yet. Nevertheless, we can not use it directly in VANET broadcast communication because some of its characteristics fail to meet some basic requirements, as is analyzed in Section 5.1. In this section we make a few modifications and enhancements of the original CL-Idemix scheme to tune it to suit the need of vehicular communication better.

Since the problems that prevent CL-Idemix from being used in VANET mainly lie in the verification protocol, the verification protocol of CL-Idemix is the main target to be considered.

First of all, the session number n_1 in verification protocol is not necessary for zero-knowledge proof. Instead, a timestamp generated by Prover is a candidate to substitute the session number. The goal of n_1 is to ensure that Prover and Verifier have a fresh connection with each other, and that the proof of the credential is specific for this connection (recall that n_1 partly controls the hash value c). In this way replay attacks are prevented. But in broadcast communication, there is no bidirectional connection between message originator and receivers. Thus n_1 is not applicable. To prevent replay attacks, we substitute a timestamp for n_1. Using a timestamp is applicable to ensure the freshness of the proof. Another benefit is it reduces the number of rounds of interaction between Prover and Verifier to be only half a round, i.e., there is no broadcast back channel, hence both single-hop broadcast authentication and multi-hop broadcast authentication become possible. All vehicles need to be time synchronized to minimize the difference of OBU local time of each other in order to use timestamp instead of n_1. As vehicles are equipped with GPS, strict time synchronization is assumed.

Secondly, the verification protocol should be modified to enable message without originator authentication. Message without originator authentication is a basic requirement of a privacy-preserving authentication scheme in VANET as the broadcast messages call for this kind of message authentication (see Chapter 2). The modification of verification protocol is introduced in Section 5.2.1.

Thirdly, the authentication information size can be reduced through prime encoding[8]. The prime encoding method is introduced by Camenisch et al. as an extension of the original CL-Idemix protocol. The number of bases (a_i) in CL signature decreases after prime encoding with a reasonable cost in computation overhead. The prime encoding method is described in Section 5.2.2 in this thesis.

Finally, as CL-Idemix does not provide anonymity revocation, optional requirements 1 and 2 are not fulfilled. In this thesis, we propose a simple anonymity revocation solution as an enhancement of CL-Idemix, which is introduced in Section 5.2.3. Optional requirement 3 ("Non-repudiation") is also enabled due to the anonymity revocation solution.

Note that in CL-Idemix, pseudonym creation and verification sub-protocols are also defined, causing a small increase of overhead to the whole system. But if we use pseudonyms, it turns into a pseudonym system scheme, rather than an attribute authentication scheme. For this concern we do not use the pseudonym functionality of CL-Idemix.

5.2.1 The Message Authentication Process

CL-Idemix does not provide message authentication because it is aimed to be used in the Internet, and there are already mature message authentication solutions for unicast communication like SSL and TLS[14]. But in vehicular com-

Msg	Timestamp	A_r	Signature	Proof

Table 5.4: Message Format

munication, broadcast of safety and traffic efficiency messages calls for message without originator authentication. For CL-Idemix, it can be transformed into an attribute authentication scheme suitable for broadcast message authentication with a small number of modifications, without security loss.

The first modification is we give a new definition of the message format. As CL-Idemix does not provide message authentication, there is no such definition of message format in CL-Idemix. The message format for VANET communication is shown in table 5.4 and explained as follow:

The "Msg" part is the message payload that is to be signed. "Timestamp" is a synchronized timestamp that Prover gets from time control applications in OBU, such as the time synchronization application of a GPS receiver. "A_r" is a list of attribute values for the revealed attributes. The "Signature" part is the message signature δ.

$$\delta = M^{m_s} \mod \text{n}$$

where M = {Msg, Timestamp}, and m_s is a hidden attribute in the CL-Idemix credential of Prover, which serves as a message signing key of Prover.

The "Proof" part contains $\{c, \mathbf{s}, \mathbf{Common}\}$ of the CL-Idemix credential verification protocol.

Message Signing

To generate a message signature that offers message without originator authentication on message Msg, Prover follows the procedure below.

1. Prover reads in values of revealed attributes A_r.
2. Prover gets a current Timestamp, combines message payload Msg and Timestamp into M.

$$M = \{\text{Msg, Timestamp}\}$$

3. Prover computes a signature δ on a M.

$$\delta = M^{m_s},$$

where m_s is a hidden attribute in the credential for Prover. m_s works as signing key in the message signing process.

4. Prover generates a random number r_{m_s} and computes t_M.

$$t_M = M^{r_{m_s}}$$

5. Prover calls CL-Idemix proof creation process, adds t_M in $\mathbf{T}$, and creates "Proof".

During the proof creation process, t_M is considered as a special t-value and is added into the t-value set $\mathbf{T}$ in CL-Idemix protocol. The created "Proof" contains $\{c, \mathbf{s}, \mathbf{Common}\}$, just as the original CL-Idemix proof. The difference is the computation of c incorporates Timestamp rather than the original n_1, and the message payload Msg is also incorporated into the computation of c.

$$c = H(context, FuncData, \mathbf{Common}, \mathbf{T}, Msg, Timestamp).$$

Note that $\mathbf{s}$ in "Proof" contains an s-value on m_s.

$$s_{m_s} = r_{m_s} + cm_s \ ,$$

6. Prover assembles {Msg, Timestamp, A_r, δ, Proof} into a message and sends out the message.

Signature Verification

After receiving a signed message {Msg, Timestamp, A_r, δ, Proof}, Verifier calls the signature verification process to verify the message signature. Verifier does the following:
1. Verifier calls CL-Idemix credential verification protocol to re-construct all t-values in $\mathbf{T}$ except t_M. $FuncData$ is also re-constructed during the execution of the protocol.
2. Verifier retrieves the element s_{m_s} from $\mathbf{s}$ and re-constructs t_M.

$$\hat{t_M} = M^{s_{m_s}} \cdot \delta^{-c}$$

Verifier adds $\hat{t_M}$ in the set $\mathbf{T}$.
3. Verifier re-constructs hash value $\hat{c}$.

$$\hat{c} = H(context, FuncData, \mathbf{Common}, \mathbf{T}, Msg, Timestamp),$$

in which *context* is a list of public parameters in CL-Idemix protocol.
4. Verifier compares $\hat{c}$ with c. If the two values are the same, Verifier decides the signature δ is valid. Otherwise Verifier decides the signature is not valid.

The whole procedure of message signing and signature verification is illustrated in Figure 5.2

Security and Correctness of the Message Authentication Process

For the correctness of the message authentication process, it depends on the correctness of $c == \hat{c}$. As the message authentication process is based on CL-Idemix, and the only change to the CL-Idemix computation of c and $\hat{c}$ is t_M, $\hat{t_M}$, and $Timestamp$ are added in the hashed data of c and $\hat{c}$, the correctness of the message authentication process depends on the correctness of $t_M == \hat{t_M}$, since $Timestamp$ is directly sent to the receiver.

$t_M == \hat{t_M}$ is correct, because this is the result of non-interactive Schnorr's Identification, and it can be validated that

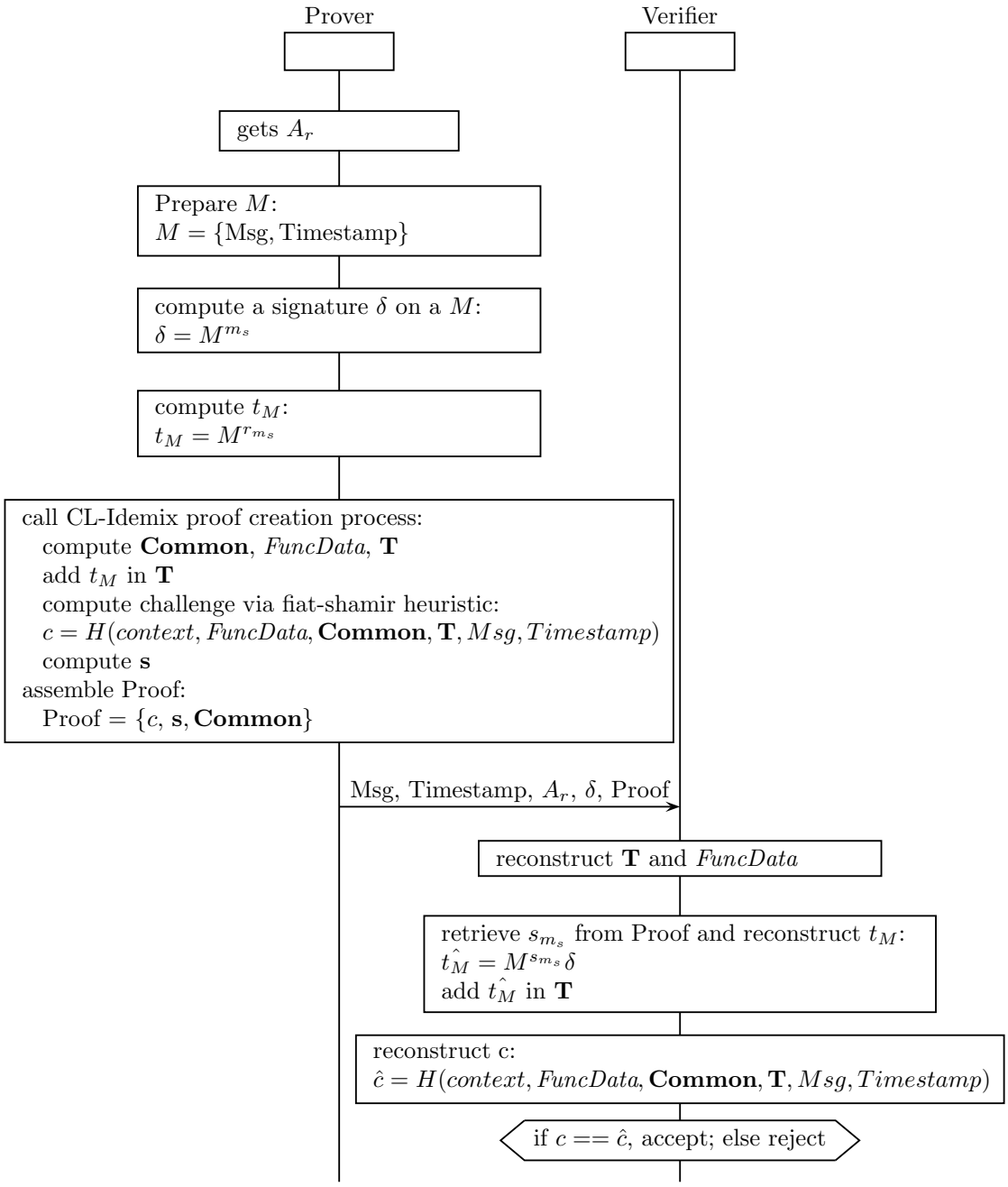

Figure 5.2: Message Signing and Signature Verification

$$
\begin{aligned}
\hat{t_M} &= M^{s_{m_s}} \cdot \delta^{-c} \\
&= M^{s_{m_s}} \cdot (M^{m_s})^{-c} \\
&= M^{r_{m_s}} \\
&= t_M
\end{aligned}
$$

For the security of the message authentications scheme, it depends on the security of non-interactive Schnorr's Identification scheme and the security of the CL-Idemix protocol suite. Here we explain the unforgeability of the message signature δ and the anonymity of the vehicle.

Unforgeability The unforgeability of the message signature δ is straightforward. If the attacker wants to forge a signature, it needs to know the value of m_s in a valid credential, as well as the other values of A_r and the credential (A, e, v). This is to say the attacker needs to forge a CL-Idemix credential before it forges a signature. The unforgeability of the CL-Idemix credential relies on the strong RSA assumption, as is introduced in Section 4.2. As a result, the unforgeability of the message signature δ also depends on the strong RSA assumption.

Anonymity As CL-Idemix verification achieves anonymous authentication (attribute authentication), it is intuitive that if the other parts of the message authentication scheme does not affect anonymity, then it is anonymity guaranteed. Since a timestamp is included in the signing message M, i.e. $M = \{Msg, Timestamp\}$, the signature $\delta = M^{m_s}$ is different when Msg is the same and $Timestamp$ is different. When Msg is not the same, δ changes anyway. So the maximum linkable time of δ for a specified vehicle is the period of $Timestamp$. For example, if the timestamp is set to change in every 1 second, the maximum linkable time is 1 second. In conclusion the maximum linkable time for a specified vehicle is the period of $Timestamp$. The message authentication scheme is not fully anonymous in this case. As the period of $Timestamp$ is usually very short, this does not incur serious anonymity problem. Nevertheless, it is certainly applicable to include extra random data, e.g., append one byte of random value, in M to make δ fully unlinkable so that the message authentication scheme is fully anonymous.

5.2.2 Prime Encoding

We use "efficient attribute" prime encoding [8] as an enhancement to CL-Idemix scheme. From Section 5.1.1, we can see that the overhead of CL-Idemix system is partly influenced by the number of attributes in the CL signature. Prime encoding encodes attribute values in prime numbers and can put the values of t attributes in a grouped attribute m of CL signature, thus reduces the number of attributes m_i in CL signature scheme. However, prime encoding only applies to

the finite set attributes, whereas the long string and big integer attributes are not prime encoded, because the values of long strings and big integers are too big to be represented by primes. In [8], the authors assume that the attributes have the same size before they are grouped, which makes the explanation easier. Nevertheless, it is straight-forward to use their method where the attributes have different sizes. In order to clarify the prime encoding process, we explain how attributes are grouped before we give the presentation of a grouped attribute m.

Grouping Attributes

Assuming the maximum length of an attribute signed in Idemix is l_m bits, and we want to group a number of attributes m_i to a new attribute m of CL-Idemix protocol. Firstly we assign ascending prime numbers to the attribute values of each attribute, starting from 2. For example, the prime numers for m_1 values are $2, 3, 5, ...p_{s_1}$, where p_{s_1} is the s_1-th prime, and s_1 is the size of the finite set for m_1. And the prime numbers for m_2 values are $p_{s_1+1}, p_{s_1+2}, ...p_{s_1+s_2}$, where s_2 is the size of the finite set for m_2. That is, for attribute m_i, we reach the x_i-th prime number p_{x_i}, where

$$x_i = \sum_{j \leq i} s_j.$$

For each i , we compute

$$m' = \prod p_{x_i}$$

and

$$L_i = bit(m'),$$

which is the big length of m', until we come to the last i that makes $L_i \leq l_m$. The rest attributes after i can not be grouped into m and should be grouped in another new attribute. If $L_i \leq l_m$ is correct for the the biggest i, then all the attributes can be grouped in m.

Representation of Prime Encoding

For a grouped attribute m, its values are represented in the way described as follows:

Again, we assign ascending prime numbers to the attribute values m_i who have been grouped into m. Suppose the prime number for the value for m_i is v_{m_i}, where

$$v_{m_i} = prime(m_i),$$

then a value of the grouped attribute m has the form of

$$v_m = \prod v_{m_i}.$$

Prime-encoded attributes do not require changes in the issuance protocol. But credential verification requires extra protocols if the attributes are prime-encoded. Readers can refer to [48] and [8] for detailed explanation of the prime-encoding enabled credential verification.

5.2.3 Anonymity Revocation

The anonymity revocation functionality is the fulfillment of optional requirement 1 and 2 ("identity resolution" and "isolation of vehicle" respectively), while at the same time it also fulfills optional requirement 3 ("non-repudiation").

Identity resolution can be achieved through making the signing key m_s a known attribute during issuance, and the CA can use the key as an identifier of the vehicle. The CA can search through its stored keys to find the one that corresponds with the specified signature of a message for a specified date. Although making the signing key a known attribute might incur the dispute that a dishonest CA may forge signature of the vehicle, it is even easier that a dishonest CA can even forge a signing key and then forge a signature. In any case a dishonest CA can cause fake signature, it is thus better to choose the previous solution to result in less number of attributes.

For the isolation of an excluded vehicle, both pre-issuing isolation and post-issuing isolation is possible. For pre-issuing isolation, a short valid period of credentials is required. In that case the vehicles update their credentials in a static frequency, e.g., once every day. Of course, the excluded vehicles should not receive an updated credential. For post-issuing isolation, a revocation list is distributed to vehicles. The revocation list contains the signing keys of the excluded vehicles. A vehicle receiving a signed message does this:

1. For each signing key m_{s_i} in the revocation list, compute the signature δ' on M.

$$\delta' = M^{m_{s_i}}$$

2. Compare δ' against δ value in the received message. If no match is found, the receiver believes the message originator is a valid vehicle.

Nevertheless, the computation of δ' increases the computation overhead of the receiver. The size of the revocation list should be less than a threshold t that would cost affordable computation time for the verifier.

5.3 System Structure and Phases of CLIBA

Having described the modified CL-Idemix protocol suite, in this section, we introduce the whole system structure and different phases of CLIBA in the background of the system structure.

5.3.1 System Structure

The system of the modified CL-Idemix scheme consists of various parts to make it function well. The system consists of a backend network and a vehicular network. The backend network is made up of the Issuer server, the intermediate transmission nodes, and the vehicles. The vehicular network is the VANET, which is made up of the vehicles. The functionality of each part of the two networks is introduced below:

Firstly, there is a centralized Issuer server storing the information for users. The stored information include the user's permanent identity number (or identity information to link to external systems), the A_k values for the users, and revocation information. The Issuer server is responsible for issuing and updating credentials for vehicles, as well as excluding misbehaving vehicles. By default, the Issuer server also performs the functionality of initializing the whole system.

Secondly, there are intermediate transmission nodes, like RSUs, cellular APs, and Wifi routers as shown in Figure 5.1. Those are typical infrastructures in VANET and can be used by our system. They can work as the agencies of the Issuer server or work as intermediate nodes between the Issuer server and the vehicles.

Finally, vehicles, or more precisely, the OBUs in vehicles, are the users of the system. In the centralized network, they request for and receive a CL-Idemix credential or update a credential from the Issuer server through the intermediate transmitters. In the decentralized VANET, they use their credentials to sign broadcasted messages. They also verify the signatures of other vehicles.

5.3.2 Phases of CLIBA

As CLIBA basically follows the steps of CL-Idemix protocol suite, the phases of CLIBA are similar to CL-Idemix. We explain the phases of the modified CL-Idemix scheme as follows:

1. **Initialization**. By default, the Issuer works as the initializer in this phase. The initializer sets system parameters and generates private keys for the CL signature. More precisely, the system parameters include the length of the parameters, the specification of the attributes, and public parameters like n, b, c, $\{a_1, a_2, ..., a_L\}$, etc. The specification of the attributes gives a list of attributes, including the name of attributes, the type of attributes, i.e., belongs to A_k or A_{UC}, and the length of the attribute values, etc. The private keys of CL signature are the safe primes p and q. To enable anonymity revocation, the signing key of vehicles is a known attribute. After initialization, the Issuer receives the private keys of CL signature, system parameters are published.

2. **Certificate Issuance and Update**. In this phase, credentials are issued and updated. After authenticating the vehicle using some third-party authentication methods such as passwords or PKI certificates (which are not a part of our attribute authentication scheme), the Issuer issues a credential (A, e, v) to the vehicle. The A_k values for the vehicle are also sent to the vehicle. The vehicle checks that the credential is correct and stores the credential and

all the attribute values (values of A_k and A_{UC}) locally. The Issuer server stores the revocation information and update information of the vehicle. When the A_k values of the vehicle change, e.g., the properties of the vehicle change, the credential is updated. The vehicle receives updated credential and A_k values in the update process.

3. **Message Signing and Verification**. When a vehicle has a valid credential and the list of attribute values for the credential, it can sign messages using one of the attribute values – the signing key m_s of the vehicle. The signing and verification processes are described in Section 5.2.1. Only the messages with a verified signature is trusted by the message receiver. If the post-issuing isolation method is used, the receivers also need to check that the message originator is not revoked.

4. **Anonymity Revocation**. When disputes happens, the law enforcement department probably wants to know the identity of the message sender. In that case, anonymity revocation is performed. The Issuer server searches its database to find the corresponding m_s of the signature, and then the identity of the message sender. If the identity of the message sender is found, the vehicle is isolated from the system, either by pre-issuing isolation or by post-issuing isolation, as introduced in Section 5.2.3.

5.4 Summary of CLIBA

The broadcast message authentication scheme described in this chapter is an enhancement of the CL-Idemix protocol suite. The scheme employs a timestamp to make the original credential verification protocol of CL-Idemix non-interactive. Also, a message authentication process is added in the credential verification protocol. Moreover, we use prime-encoding to reduce the authentication information size. Finally, an extra anonymity revocation process for CL-Idemix credentials is added in this scheme. In next chapter, we evaluate our scheme according to the requirements listed in Chapter 2.

Chapter 6

Evaluation and Analysis

In Chapter 2, we list the requirements for a secure and privacy-preserving broadcast authentication scheme. Consequently, we evaluate the related schemes according to the specific requirements in Chapter 3. In this chapter, we evaluate the CLIBA(CL-Idemix based Broadcast Authentication scheme) that we described in Chapter 5 according to the requirements. We also present and discuss the benchmark result in this chapter.

6.1 Evaluation on CLIBA

Firstly, we give the evaluation of all the requirements except the information size and computation overhead, as they need experimental test.

For the basic requirements,

1. Message Authentication Without Originator Verification. CLIBA fulfills this requirement since the underlying Cl-Idemix protocols are based on zero-knowledge proof. The originator identification information is not revealed during the authentication if the prover does not reveal it intentionally.

2. Attribute Authentication. CLIBA is an attribute authentication scheme inherently, as the CL-Idemix protocols realize attribute authentication.

3. Privacy Protection. CLIBA provides full anonymity for the message sender (see Section 5.2.1), and thus does not leak privacy infringing information about the vehicle.

4. Strong Unlinkability. Due to the anonymity characteristic, the linkable time of a vehicle is 0 if any two broadcasted messages take different timestamps, or if extra random data is included in the message. One may seek attacks based on driving behavior and geographical conditions to predict the movement of vehicles. In that case the "mix zone" method (c.f. Section 3.1.5) is a possible solution.

5. One-hop Broadcast Authentication. As we modified CL-Idemix to make the transmission of the authentication information one-way, CLIBA is a non-interactive authentication scheme, thus fulfills one-hop broadcast authentication.

6. Small Size. The evaluation of the information size and computation overhead is shown in Section 6.1.1.

7. Low Computation Overhead. See Section 6.1.1.

8. Independent Authentication. After receiving the credential, the vehicle can use it for long-term purpose, until the credential expires or is revoked.

For the optional requirements:

1. Resolution of anonymity. Resolution of anonymity can be achieved if the signing key is a known attribute to the CA, which is described in Section 5.2.3.

2. Isolation of Vehicle. The anonymity revocation process in Section 5.2.3 provides two isolation methods. Using either method, this requirement can be fulfilled. However, the post-issuing revocation method will cause heavier computation effort at vehicle side, whereas the pre-issuing revocation method will make the valid time of the credential short.

3. Non-repudiation. Due to the unforgeability of the message signature, the non-repudiation is fulfilled, although it is based on the assumption that the CA is honest. We discuss the honesty of CA and its influence to the security of the system in Chapter 7.

4. Sybil Attack Suppression. CLIBA does not prevent Sybil attack. To some extent it even helps Sybill attack because of the extreme high unlinkability degree of the messages. In order to prevent Sybil attack, the number of messages a vehicle can sign at a certain time should be restricted, which is not guaranteed by CLIBA.

5. Multi-hop Authentication. The scheme can be used in multi-hop scenarios as the messages are completely uni-directional and there is no back channel at all.

6.1.1 Experiments and the Results

We implemented CLIBA in C++ based on the crypto++ library[1]. We mainly used crypto++ library to generate random numbers and to perform computation tasks including modular arithmetic and operations on large integers. The benchmark is performed on a PC with Ubuntu 10 system. The PC has a Intel Core i3 CPU with a frequency of 2.13 GHz, and 4GB RAM.

[1] www.cryptopp.com

Attribute Name	Type	Bitlength /Finite Set Size
signingKey	big integer	224 bit
vehicleDistrict	finite set	23
vehicleRole	finite set	5
vehicleType	finite set	15
validYear	finite set	5
validMonth	finite set	12
validDate	finite set	31

Table 6.1: Attribute List For Scenario 1

The program consists of the initialization, the credential issuance, and the signing and verification process. For simplicity, we only implemented the mandatory protocols in Idemix. In addition, we also implemented a pair of the subprotocols of "Prove Prime Encoding" protocol and "Verify Prime Encoding" protocol: "ProveCGAND" and "VerifyCGAND", because they can prove (or verify) multiple prime encoded attributes at a time, e.g., a vehicle being a private car. We prepare two scenarios to test the program. One scenario models the simple situation where only a small set of attributes are included in the attribute list of the credential. The attributes are divided into two subsets, i.e., the known attributes A_k and hidden attributes A_h. The attributes are

$$\{role, type, district, valid\,year, valid\,month, valid\,date\} \subset A_k$$

$$\{signing\,key\} \subset A_h$$

The sizes of the attributes in this scenario is shown in Table 6.1. The attributes "validYear", "validMonth" and "validDate" altogether form the expiration date of the vehicle. Here "validYear" is the number of years after the initialization time of the system. We assume the system to be re-initialized in 5 years in this scenario, that is, the system keys and parameters will be refreshed in 5 years. In actual usage, the life of the system keys can be even shorter if the time for cracking the system keys gets shorter. And the expiration date of all credentials should be the same to avoid tracking by the expiration date. The signing key of vehicles has a bit length of 224 bits. We choose this bit length according to the recommendation of key length from NIST [31], which specifies a ">= 224 bit" length for finite field discrete logarithm based keys in the years between 2011 and 2030.

The other scenario models the situation where there are more attributes, some of which are big finite attributes that can be grouped using prime encoding. Doing this enables us to see the difference on the computation overhead between the two scenarios. The sizes of attributes in this scenario are listed in Table 6.2.

In both scenarios, we group the finite attributes with prime encoding. We restrict the number of encoded attribute values t in one message field of CL

Attribute Name	Type	Bitlength /String Length /Finite Set Size
signingKey	big integer	224 bit
vehicleDistrict	finite set	23
vehicleRole	finite set	5
vehicleType	finite set	15
validYear	finite set	5
validMonth	finite set	12
validDate	finite set	31
testStrAttr1	string	20 characters
testStrAttr2	string	100 characters
testIntAttr1	integer	12345 bit
testFinAttr1	finite set	654321
testFinAttr2	finite set	999999
testFinAttr3	finite set	34212
testFinAttr4	finite set	4567
testFinAttr5	finite set	777
testFinAttr6	finite set	485020
testFinAttr7	finite set	9381

Table 6.2: Attribute List For Scenario 2

Scenario	Process	Average	Standard Deviation
1	Sign	64.5 ms	1.1ms
1	Verify	45.8 ms	1.0ms
2	Sign	71.5 ms	1.9 ms
2	Verify	52.4 ms	0.25 ms

Table 6.3: Real Time Consumption

signature to be 1,000,000. The reason to do this is the vehicle is supposed to have a record of all possible prime numbers until the t-th prime. If t is big, then the memory usage is also high. It is not efficient if vehicles calculate the primes at run time. Thus, all the finite attributes in scenario 1 can be encoded into one message field of CL signature, since the total number of finite attribute values in that scenario is 91, which is far less than t. Finally we get 2 encoded message fields in the first scenario, and 7 encoded message fields in the second scenario.

We measured the running time of the signing and verification process by setting inline timers in the program. More precisely, we used the function gettimeofday() to get the current time at the beginning and end point of the process, and calculated the time interval. The advantage of gettimeofday() is it has a precision of 1 microsecond, which is good enough for this benchmark. The disadvantage is it can only get real time record and thus would underestimate the speed of the program. To avoid the disadvantage, we tested the program with highest priority in protection mode of Ubuntu, i.e., there is no interference from other programs. We run the broadcast authentication system for the two scenarios and tested 100 rounds of the signing and verification process. The average real time consumption is shown in Table 6.3. We can see that a small increase appears in the second scenario compared with the first scenario, which is due to the greater number of attributes.

It is worth mentioning that the time for multi-exponentiation calculation counts 85% of the computation time, because the multi-exponentiation algorithm that we use is not efficient enough. The time for each step in a typical verification run is shown in Table 6.4, in which hatC_C0 and CascadeExp_baelists are the multi-exponentiation steps.

In addition to the heavy computation overhead, the information size is also big. When revealing 2 finite attributes, scenario 1 has an authentication information size of around 1800B. When revealing 3 finite attributes and 2 integer and string attributes, scenario 2 has an authentication information size of 2070B. This is because the modulus n of CL-Idemix protocols has a bit length of 2048 bit (c.f. Table 5.3), which result in large information size based on n,

It can be thus concluded that the computation overhead is generally acceptable for an 100 ms schedule for outgoing messages. However for incoming messages, the verification time is too long for vehicular broadcast as there are always more message for verification than for signing. This problem may be

Step	Time (in µs)	Multi-Exponentiation	Proportion
parsing	92	N	0.20%
hatC_C0	18254	Y	40.28%
CascadeExp_baelists	20436	Y	45.09%
signing	4137	N	9.13%
other steps	2401	N	5.30%
all steps	45319	N	–

Table 6.4: Real Time Consumption of Each Steps

Req.1	Req.2	Req.3	Unlinkability	Req.5	Size	Time (verification)	Req.8
Y	Y	Y	High	Y	Very Big	Very Big	Y

Table 6.5: The Fulfillment of Basic Requirements for CLIBA

relieved by hardware acceleration and faster multi-exponentiation algorithms. The authentication information size is really big and is not applicable for vehicular broadcast.

6.2 Summary of the Evaluation Result

An overview of the evaluation result for the basic requirements is shown in Table 6.5, and the evaluation result for the optional requirements is shown in Table 6.6. CLIBA satisfies most of the basic requirements except that the overhead is too big. The computation time can only meet the outgoing requirement, i.e., message signing. As most of the computation time is spent on multi-exponentiation, a more efficient multi-exponentiation algorithm is needed to reduce the computation overhead. Usable algorithms are described in [30]. For the optional requirements, CLIBA also satisfies most requirements except that it does not prevent Sybil attacks. Indeed, the extreme high unlinkability degree even makes the Sybil attacks easier.

Req.1	Req.2	Req.3	Req.4	Req.5
Y	Y	Y	N	Y

Table 6.6: The Fulfillment of Basic Requirements for CLIBA

Chapter 7

Conclusion and Future Work

In VANET environment, privacy protection is a big issue. Many authentication schemes were brought out to protect privacy of vehicles, e.g., pseudonym systems and group signing. However, those schemes does not provide attribute authentication. It is the goal of this thesis to seek a way to apply attribute authentication in VANET. CL-Idemix is an attribute authentication protocol suite for Internet environment, which we want to adapt to VANET. As CL-Idemix is meant to be used in Internet, some of its characteristics do not qualify the usage in VANET. For example, the verification protocol is interactive, there is no integrated message authentication with the attribute authentication scheme, and no anonymity revocation.

CLIBA is a CL-Idemix adaptation with a new message authentication mechanism accompanied with the attribute authentication scheme of CL-Idemix. It is non-interactive in verification. Besides, there are anonymity revocation methods for CLIBA. It satisfies most of the requirements of a secure and privacy-preserving broadcast authentication scheme, which makes it a valid candidate for being a broadcast authentications scheme in VANET. But there are still obstacles that prevent CLIBA from being used, that is its slow verification speed and big authentication information size. Moreover, it does not prevent Sybil attack, due to its extreme high unlinkability degree of messages for the vehicle. Therefore, a light-weight broadcast authentication scheme that can remedy the disadvantages of CLIBA is needed. The open issues to be discussed in the future are listed in the below:

1. Verification speed up. Hardware acceleration and multi-exponentiation algorithm enhancement are possible measurement for the problem. There are common-purpose hardware accelerators for asymmetric cryptographic usage, such as SSL accelerators[1]. SeVeCom also suggests using hardware

[1] http://sslacceleration.info/

security modules for vehicles[25]. For CLIBA, multi-exponentiation computation is the most time-consuming operation. Either a hardware acceleration module of multi-exponentiation or a faster algorithm can solve the problem. Various fast multi-exponentiation algorithms are described in [30].

2. Authentication information size reduction. The authentication information size is too big to be applicable in VANET. As CL-Idemix is based on strong RSA assumption, the information size is inherently large. This means we need to bypass the RSA based cryptographic systems in order to get compact signatures and proofs.

3. Sybil attack prevention. CLIBA has an extreme high unlinkability in messages sent by the vehicle, which even helps launching a Sybil attack. A Sybil attack prevention mechanism is required in future work. Either Sybil attack is avoided, or it is detected after the attack is launched. Camenisch et. al[9] introduced a Sybil attack prevention scheme which also uses CL signature as a building block. In their scheme the number of messages that a vehicle can sign in a solid time period is restricted to a number n. If a vehicle signs more than n messages in a time period, it will face the danger of identity exposure. As their scheme also bases on CL signature, there is a possibility to modify it to fit VANET communication.

4. CA honesty guarantee. When anonymity revocation is enabled, the CA is trusted to identify the originator of a signed message, as the CA has a record of all the signing keys of vehicles. One might worry that the CA is also capable to forge a signature for a vehicle if it has the signing key. But consider that if a CA is dishonest, it can not only forge a signature based on the recorded key, it can also forge a signature using a fresh new signing key and assign this key to the specified vehicle. Even if the CA has no record of signing keys, it can forge a signing key for the vehicle since nobody knows which is the correct signing key. Hence a mechanism that can prevent CA from forging signatures is also required.

To sum it up, the CL-Idemix based Broadcast Authentication scheme can be adapted to VANET broadcast communication in the cost of high performance requirement and large authentication information size. Other issues like Sybil attack and CA dishonesty should also be addressed in the future to make CLIBA more secure.

Bibliography

[1] C.A. Ardagna, J. Camenisch, M. Kohlweiss, R. Leenes, G. Neven, B. Priem, P. Samarati, D. Sommer, and M. Verdicchio. Exploiting cryptography for privacy-enhanced access control: A result of the prime project. *Journal of Computer Security*, 18(1):123–160, 2010.

[2] F. Armknecht, A. Festag, D. Westhoff, and K. Zeng. Cross-layer privacy enhancement and non-repudiation in vehicular communication. In *4th Workshop on Mobile Ad-Hoc Networks (WMAN)*. Citeseer, 2007.

[3] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[4] P. Bichsel, C. Binding, J. Camenisch, T. Groß, T. Heydt-Benjamin, D. Sommer, and G. Zaverucha. Cryptographic protocols of the identity mixer library. 2009.

[5] Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In *Proceedings of the 11th ACM conference on Computer and communications security*, CCS '04, pages 168–177, New York, NY, USA, 2004. ACM.

[6] Levente Buttyán, Tamás Holczer, and István Vajda. On the effectiveness of changing pseudonyms to provide location privacy in vanets. In *Proceedings of the 4th European conference on Security and privacy in ad-hoc and sensor networks*, ESAS'07, pages 129–141, Berlin, Heidelberg, 2007. Springer-Verlag.

[7] Giorgio Calandriello, Panos Papadimitratos, Jean-Pierre Hubaux, and Antonio Lioy. On the performance of secure vehicular communication systems. *IEEE Transactions on Dependable and Secure Computing*, 99(PrePrints), 2010.

[8] Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials. In *Proceedings of the 15th ACM conference on Computer and communications security*, CCS '08, pages 345–356, New York, NY, USA, 2008. ACM.

[9] Jan Camenisch, Susan Hohenberger, Markulf Kohlweiss, Anna Lysyanskaya, and Mira Meyerovich. How to win the clonewars: efficient periodic n-times anonymous authentication. In *Proceedings of the 13th ACM conference on Computer and communications security*, CCS '06, pages 201–210, New York, NY, USA, 2006. ACM.

[10] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Giuseppe Persiano, and Clemente Galdi, editors, *Security in Communication Networks*, volume 2576 of *Lecture Notes in Computer Science*, pages 268–289. Springer Berlin / Heidelberg, 2003.

[11] The CAMP Vehicle Safety Communications Consortium. Vehicle safety communications project task 3 final report identify intelligent vehicle safety applications enabled by dsrc. Technical Report DOT HS 809 859, National Highway Traffic Safety Administration U. S. Department of Transportation (USDOT), March 2005.

[12] D. Cooper. Internet x.509 public key infrastructure certificate and certification revocation list (CRL) profile. *RFC 5280*, 2008.

[13] Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In *ASIACRYPT*, pages 125–142, 2002.

[14] T. Dierks. The transport layer security (tls) protocol. *RFC 5246*, 2008.

[15] Florian Dötzer. Privacy issues in vehicular ad hoc networks. In George Danezis and David Martin, editors, *Privacy Enhancing Technologies*, volume 3856 of *Lecture Notes in Computer Science*, pages 197–209. Springer Berlin / Heidelberg, 2006.

[16] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[17] M. Emmelmann, B. Bochow, and C. Kellum. *Vehicular Networking: Automotive Applications and Beyond*. Intelligent Transport Systems. Wiley, 2010.

[18] L. Fischer, A. Aijaz, C. Eckert, and D. Vogt. Secure revocable anonymous authenticated inter-vehicle communication (sraac). In *4th Conference on Embedded Security in Cars (ESCAR 2006), Berlin, Germany*. Citeseer, 2006.

[19] Julien Freudiger, Maxim Raya, Mrk Flegyhzi, Panos Papadimitratos, and Jean-Pierre Hubaux. Mix-Zones for Location Privacy in Vehicular Networks. In *ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS)*, Vancouver, 2007.

[20] H. Hartenstein and K. Laberteaux. *VANET: vehicular applications and inter-networking technologies*. John Wiley & Sons Inc, 2010.

[21] Florian Hess. Efficient identity based signature schemes based on pairings. In *SAC 2002, LNCS 2595*, pages 310–324. Springer-Verlag, 2002.

[22] Pandurang Kamat, Arati Baliga, and Wade Trappe. An identity-based security framework for vanets. In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, VANET '06, pages 94–95, New York, NY, USA, 2006. ACM.

[23] F. Kargl, Z. Ma, and E. Schoch. Security engineering for vanets. *Proc. 4th Wksp. Embedded Sec. in Cars*, pages 15–22, Nov. 2006.

[24] F. Kargl, P. Papadimitratos, L. Buttyan, M. Muter, E. Schoch, B. Wiedersheim, Ta-Vinh Thong, G. Calandriello, A. Held, A. Kung, and J.-P. Hubaux. Secure vehicular communication systems: implementation, performance, and research challenges. *IEEE Communications Magazine*, 46(11):110–118, 2008.

[25] Frank Kargl, Panos Papadimitratos, Levente Buttyan, Michael Müter, Björn Wiedersheim, Elmar Schoch, Ta-Vinh Tongh, Giorgio Calandriello, Albert Held, Antonio Kung, and Jean-Pierre Hubaux. Secure Vehicular Communications: Implementation, Performance, and Research Challenges. *IEEE Communications Magazine*, 46(11):2–8, November 2008.

[26] X. Lin, X. Sun, P. . Ho, and X. Shen. Gsis: A secure and privacy-preserving protocol for vehicular communications. *IEEE Transactions on Vehicular Technology*, 56(6 I):3442–3456, 2007. Cited By (since 1996): 103.

[27] R. Lu, X. Lin, H. Zhu, P.H. Ho, and X. Shen. Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 1229–1237. IEEE, 2008.

[28] A.J. Menezes, P.C.V. Oorschot, and S.A. Vanstone. *Handbook of applied cryptography*. CRC Press series on discrete mathematics and its applications. CRC Press, 1997.

[29] IDM Modinis. Common terminological framework for interoperable electronic identity management, November 2005.

[30] Bodo Möller. Algorithms for multi-exponentiation. In *Revised Papers from the 8th Annual International Workshop on Selected Areas in Cryptography*, SAC '01, pages 165–180, London, UK, UK, 2001. Springer-Verlag.

[31] NIST. Recommendation for key management, special publication 800-57 part 1 rev. 3, 05 2011.

[32] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Zhendong Ma, F. Kargl, A. Kung, and J.-P. Hubaux. Secure vehicular communication systems: design and architecture. *IEEE Communications Magazine*, 46(11):100–109, 2008.

[33] Panagiotis (Panos) Papadimitratos, Ghita Mezzour, and Jean-Pierre Hubaux. Certificate revocation list distribution in vehicular communication systems. In *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*, VANET '08, pages 86–87, New York, NY, USA, 2008. ACM.

[34] Adrian Perrig, Ran Canetti, J.D̃. Tygar, and Dawn Song. The tesla broadcast authentication protocol. *RSA CryptoBytes*, 5(Summer), 2002.

[35] Florian Schaub, Frank Kargl, Zhendong Ma, and Michael Weber. V-tokens for conditional pseudonymity in vanets. In *IEEE Wireless Communications & Networking Conference (IEEE WCNC 2010)*, Sydney, Australia, 04/2010 2010. IEEE, IEEE.

[36] Florian Schaub, Zhendong Ma, and Frank Kargl. Privacy requirements in vehicular communication systems. In *IEEE International Conference on Privacy, Security, Risk, and Trust (PASSAT 2009), Symposium on Secure Computing (SecureCom09)*, Vancouver, Canada, 08/2009 2009.

[37] C. Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *Advances in Cryptology CRYPTO 89 Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer Berlin / Heidelberg, 1990.

[38] A. Studer, F. Bai, B. Bellur, and A. Perrig. Flexible, extensible, and efficient vanet authentication. *Journal of Communications and Networks*, 11(6):574–588, 2009. Cited By (since 1996): 2.

[39] Jinyuan Sun, Chi Zhang, and Yuguang Fang. An id-based framework achieving privacy and non-repudiation in vehicular ad hoc networks. In *Proc. IEEE Military Communications Conf. MILCOM 2007*, pages 1–7, 2007.

[40] T. Van Deursen, S. Mauw, and S. Radomirović. Untraceability of rfid protocols. In *Proceedings of the 2nd IFIP WG 11.2 international conference on Information security theory and practices: smart devices, convergence and next generation networks*, pages 1–15. Springer-Verlag, 2008.

[41] T. van Deursen and S. Radomirovic. Attacks on rfid protocols. Cryptology ePrint Archive, Report 2008/310, 2008. `http://eprint.iacr.org/`.

[42] A.F. Westin. *Privacy and freedom.* Atheneum, 1970.

[43] ETSI ITS WG1. Intelligent transport systems (its); security; threat, vulnerability and risk analysis(tvra), March 2010.

[44] ETSI ITS WG1. Intelligent transport systems (its); vehicular communications; basic set of applications; part 1: Functional requirements, September 2010.

[45] ETSI ITS WG1. Intelligent transport systems (its); vehicular communications; basic set of applications; part 2: Specification of cooperative awareness basic service, April 2010.

[46] S.I.L.T. Workshop. *Privacy in the Digital Environment*. Haifa Center of Law & Technology, 2005.

[47] Ke Zeng. Pseudonymous PKI for ubiquitous computing. In Atzeni, AS and Lioy, A, editor, *PUBLIC KEY INFRASTRUCTURE, PROCEEDINGS*, volume 4043 of *LECTURE NOTES IN COMPUTER SCIENCE*, pages 207–222, HEIDELBERGER PLATZ 3, D-14197 BERLIN, GERMANY, 2006. Ist Super Mario Boella, SPRINGER-VERLAG BERLIN. 3rd European Public Key Infrastructure Workshop (EuroPKI 2006), Turin, ITALY, JUN 19-20, 2006.

[48] IBM Research Zurich. Specification of the identity mixer cryptographic library. `http://www.zurich.ibm.com/~pbi/identityMixer_gettingStarted/ProtocolSpecification_2-3-2.pdf`, 12, 2010.