electronics

MDPI

Article

Verification of an Evolving Security Scheme in the Internet of Vehicles

Weiyang Xin [1], Hanning Zhang [2], Chunxiang Xu [1], Jiangyuan Yao [1,*], Deshun Li [1] and Ting Jin [1]

[1] School of Computer Science and Technology, Hainan University, Haikou 570228, China; xwy@hainanu.edu.cn (W.X.); 20085400210080@hainanu.edu.cn (C.X.); lideshun@hainanu.edu.cn (D.L.); jinting@hainanu.edu.cn (T.J.)
[2] China Unicom (Hainan) Innovation Research Institute, Hainan Branch, China Unicom, Haikou 570100, China; zhanghn@stu.xjtu.edu.cn
* Correspondence: yaojy@hainanu.edu.cn

Abstract: The security scheme of the Internet of Vehicles (IoV) protects the vehicles from network attacks. However, during the experimental deployment of the IoV, people usually pay attention to its function and performance, and only develop a security scheme empirically when security defects are found. When a security scheme becomes very complex, it is very difficult to find the security vulnerability and to modify it. In view of this, we propose a verification method to verify the security of an evolving security scheme. This method uses formal methods to verify the evolving security scheme, actively finds the security problems of the security scheme, and promotes the evolution of the security scheme accordingly. This method is applied to the scenario of the IoV and its security scheme—the method evolves the Internet of Vehicles configuration and the security properties, establishes a corresponding formal model, and then iteratively verifies this using a formal method. The approach can fully simulate the evolution process of a security scheme in the IoV during deployment, and can effectively find the corresponding security vulnerabilities, to promote the evolving security scheme in the IoV, which supports the feasibility and usability of the method.

Keywords: Internet of Vehicles; evolving security scheme; security property

Citation: Xin, W.; Zhang, H.; Xu, C.; Yao, J.; Li, D.; Jin, T. Verification of an Evolving Security Scheme in the Internet of Vehicles. *Electronics* **2023**, *12*, 4438. https://doi.org/10.3390/electronics12214438

Academic Editor: Kah Phooi Seng

Received: 26 September 2023
Revised: 20 October 2023
Accepted: 26 October 2023
Published: 28 October 2023

1. Introduction

The Internet of Vehicles (IoV) [1–4] refers to the integration of automobiles with the Internet, cloud computing, and other emerging technologies. The IoV is a specific application of the Internet of Things that uses multiple communication technologies to connect vehicles to other vehicles, infrastructure, and the Internet, enabling them to communicate with each other and with their surroundings. This communication allows vehicles to share information, such as location, speed, road conditions and traffic patterns, enabling vehicles to make informed decisions and to take appropriate actions [5,6]. Integrating the IoV into an intelligent transportation system can improve road safety and traffic efficiency, manage traffic congestion more conveniently and quickly, and give drivers a better driving experience.

The development of technologies such as 5G, soft-defined networks (SDNs) [7], and multi-access edge computing (MEC) [8] provides unlimited possibilities for the rise of the IoV. The technologies of 5G and device-device (D2D) provide support for the provision of ultra-low latency, ultra-wide bandwidth, and ultra-high reliability vehicle networking [9]. SDNs can support the dynamic characteristics of VANET and ITS applications to realize the virtualization of wireless resources to promote large-scale network management and optimization [9]. Since the edge servers are close to the mobile users, MEC has inspired many new applications in the IoV, such as driver identification, real-time traffic estimation, and public safety. These applications can promote the intelligence of the IoV [10].

In the IoV, as for other technologies, there are many security vulnerabilities [11] and threats [12] related to vehicle networking. The fast-paced nature of vehicles and network

interconnections means that any security vulnerabilities or attacks can have direct and serious consequences. For example, malicious attackers can use moving picture experts group audio layer III (MP3) files to infect the entire network [13] or exploit vulnerabilities to manipulate cars, which could seriously affect the safety of passengers and pedestrians. Therefore, it is crucial to establish a strong security scheme to protect the integrity and security of the vehicle networking ecosystem. Currently, many researchers are dedicated to studying vehicle networking security schemes and have achieved a great deal [14–17]. However, in the actual deployment of vehicle networking, the initially designed security scheme may be relatively simple, so that people usually focus on its practicability and ignore the security, creating security vulnerabilities. When the security defects are exposed, the security scheme is evolved only passively and empirically. At the same time, as the deployment scenario evolves from simple to complex, the network scale evolves from small to large, the network nodes evolve from few to many, with exposure of security vulnerabilities, and the security scheme is constantly adjusted and gradually improved during deployment. The level of security protection is improved while also increasing the cost of the changes made. Therefore, how to more actively and efficiently promote the evolution of a security scheme and to identify security issues, how to design a more complete and secure security scheme before deploying it, and how to adapt to evolving application scenarios with lower change costs in subsequent deployments are important issues facing network operators.

To effectively ensure the completeness and safety of a security scheme in evolving application scenarios, this paper proposes a verification method for evolving security schemes based on a formal method and applies this method to verify the evolving security scheme of the IoV. In this method, firstly, we model the initial version of the security scheme, and extract the security properties from the security scheme. Then, we use the formal tool ProVerif [18] to verify the model. We also use the method of incremental verification when the security properties undergo change. The verification results using this method provide positive feedback to inform the evolution of the security scheme. That is, according to two different verification results (with counter-examples and without counter-examples), different verification processes are introduced. The different verification processes promote the evolution and improvement of the security scheme; then, verification is performed again. This iterative verification process is repeated until there are no counter-examples in the verification results and no configuration updates in the deployment scenario. At this point, a complete and secure final version of the security scheme is produced that meets the safety requirements. We apply this method to the IoV scenario, verifying the security of the evolving security scheme.

The major contributions of this paper are summarized in the following points:

- Description of a verification method for evolving security schemes with feedback mechanisms, and implementation of this method to verify the security scheme according to the evolution of the vehicle network scenario and the security scheme to which it is applied. The continuous evolution of the security scheme is enabled to produce a final version of the security scheme with integrity and security that can be deployed in the IoV, which reduces the cost of changes during the actual deployment of the security scheme.
- Based on formal methods for evolving a vehicle networking security scheme and determining the security properties, formal models are established and the formal verification tool, ProVerif, is used for security scheme verification.
- Application of incremental verification methods to verify the security properties to reduce the resource overhead during verification.

The remainder of this paper is organized as follows: Related works are listed in Section 2. We describe our verification method in Section 3. Then, we apply our approach to the IoV scenario in Section 4. Finally, we summarize our current and future work in Section 5.

2. Related Works

Zhang et al. [19] conducted a formal analysis of the 5G extensible authentication protocol transport layer security (5G EAP-TLS) protocol and evolved the first pi-calculus model for 5G EAP-TLS. They successfully identified two security vulnerabilities in 5G EAP-TLS, namely, the man-in-the-middle attack and the authentication between the home network side and the user. Based on the verification results, Zhang et al. proposed an improvement to the 5G EAP-TLS protocol, which satisfies the ability to resist man-in-the-middle attacks, but still lacks the ability to resist DoS attacks. Research on the security analysis of 5g protocols also includes [20–22]. Cas and Martin conducted a fine-grained formal analysis on the components of the 5G authentication and key agreement (5G-AKA) in [21], and the authors of [20,22,23] used different formal tools to analyze the 5G-AKA protocol to perform a comprehensive system evaluation. Feng et al. [24,25] formally analyzed the security assumptions and security goals required for different scenarios of the fast identity online universal authentication framework (FIDO UAF) protocol, and established the minimum security assumptions required to achieve each security goal. Based on the FIDO UAF model, authenticator rebinding attacks were found on the two apps of JD Finance and Baopay.

Bhargavan et al. [26] modeled and analyzed the TLS 1.3 draft 18 protocol using ProVerif and CryptoVerif, respectively, revealing some loopholes in the protocol draft, and proposed the reference implementation RefTLS of TLS 1.3. Cremers et al. [27] constructed a more standardized fully annotated version protocol model for TLS1.3 draft 21.

The security schemes of the above verification approaches were not specifically aimed at the security schemes in the Internet of Vehicles. In the following, research on the verification of security schemes in the Internet of Vehicles is introduced

In order to achieve secure communication between various devices in the IoV, Wazid et al. [16] designed a secure authentication key management security scheme called AKM-IoV in a fog-computing-based IoV deployment. They then used the automated validation of Internet security protocols and applications (AVISPAs) to formalize its security verification. However, Saleem et al. [28] found that the scheme was vulnerable to vehicle impersonation, fog server impersonation, RSU impersonation, and cloud server impersonation attacks. Bojjagani et al. [29] used Scyther and Tamarin to verify the security of a system named the authentication and key management protocol IoV (AKAP-IoV). The system supports secure communication, mutual authentication, and key management of the various components of the IoV. Furthermore, Bojjagani et al. logically evaluated the security properties of the system using a real or random oracle model. Seeking to address the challenging issues of security and privacy in vehicular ad hoc networks (VANET), Al-Shareeda et al. [30] proposed a privacy-preserving communication scheme based on VANET, and adopted the Burrows–Abadi–Needham (BAN) logic and random oracle model to analyze it. According to the analysis results, the scheme satisfied security and privacy, and could resist the impact of attacks, such as replay attacks, impersonation attacks, and man-in-the-middle attacks. Zelle et al. [31] used Tamarin to perform a security analysis on the Scalableservice-Oriented MiddlewarE over IP (SOME/IP), identifying three different types of man-in-the-middle attacks, and proposed two security solutions. Vasudev et al. [17] designed a lightweight IoV communication protocol, which allows the vehicle and server to establish a key, which is used for vehicle-to-vehicle (V2V) secure communication.

Most of the existing research, whether concerning the security scheme for the Internet of Vehicles or the non-Internet of Vehicles, is aimed at a single security scheme that does not change, and does not consider the verification of the evolving security scheme. However, the use of an effective verification method for an evolving security scheme can save a lot of verification costs and reduce the complexity of verification when the security scheme version is iterated. Therefore, it is necessary to propose a security scheme verification method suitable for the evolution of a security scheme.

3. Methodology

In this section, we present our verification method in detail, corresponding to Figure 1. In Section 3.1, we formally describe the security scheme and the security properties. In Section 3.2, we formally describe different update processes for the evolving security scheme. In Section 3.3, we describe the verification process step-by-step.

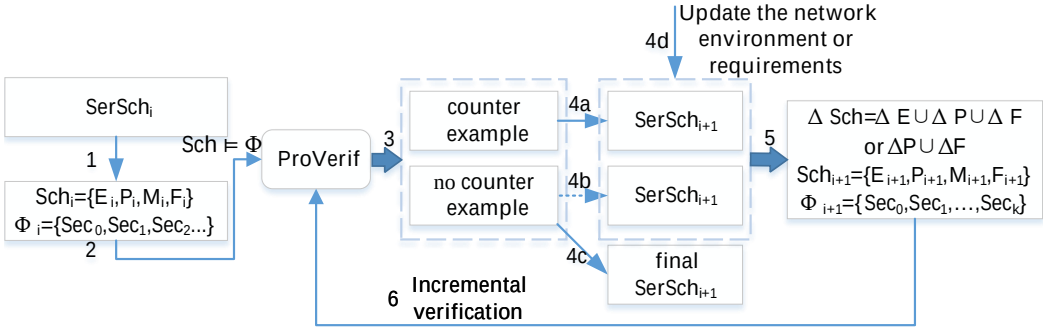

Figure 1. Verification method of evolving security schemes.

3.1. Security Schemes

In an evolving security scheme, improper network configuration may affect the security performance, and the wrong network configuration may even affect data transmission. The verification of the evolving security scheme is mainly aimed at the network configuration.

In order to better describe the evolution process of a security scheme, it is assumed that the security scheme is a quadruple $Sch = \{E, M, P, F\}$, as shown in Figure 1; Sch represents the abstract expressions extracted from the security scheme. The meaning of each element of Sch is as follows:

(1) $E = \{e_1, e_2, \ldots, e_n\}$ is the set of entities of the security scheme. The entities of the security scheme can be network users, vehicle devices, or specific institutions, such as a certificate authority. They can also be servers, switches, or other network-connected devices.

(2) P is a set of logical paths, and each path p_{ij} is represented by the entities e_i and e_j, which have a logical path for transmitting messages, which may be composed of multiple physical links.

(3) M is a set of message structures, and each message structure may consist of the source port, the destination port, the data packet, the source IP address, the destination IP address, and the data content.

(4) F is a set of message transmission processes. Each message transmission process $f_m = send(e_i, e_j, m_k)$ is a function of the entities and messages. It describes the process of entities exchanging information according to a certain order and rules. It is responsible for message distribution, expiration processing, message update, etc. Here, e_i and e_j represent the sender and receiver of the message, respectively, and m_k represents the transmitted message.

The security properties represent the security requirements—the security features that the security scheme needs to meet. In the data model of the current security scheme, Sch, during the verification process, the security properties are abstracted as $\Phi = \{Sec_0, Sec_1, Sec_2, \ldots, Sec_k\}$, as shown in Figure 1. Each Sec_k represents a specific security property, such as the reachability, confidentiality, mutual authentication, and so on.

The security verification of the evolving security scheme can be formally expressed as:

$$Sch \vDash \Phi \tag{1}$$

It is determined whether Sch satisfies the security property Φ.

3.2. Evolving Deployment

During the deployment phase of the security scheme, the message format of the general security scheme remains unchanged, and the network configuration will continue to be updated with update of the security scheme. This means that M remains unchanged, while the E, P, and F undergo changes. Therefore, the verification of the security scheme mainly focuses on the update of the E, P, and F.

As shown in Figure 1, the update process of the evolving security scheme may concern the following three situations:

- The updating of the E leads to the updating of the P. After the initial version $SerSch_0$ of the security scheme is verified, it evolves to a network environment with more entities. In the next version, $SerSch_1$, it is updated on the basis of E_0, forming a new entity set, E_1. Since the update of the network entities will lead to path changes, the P_0 undergoes changes represented by ΔP, forming the new path set. Additionally, the increase in entities will lead to communication between the increased entities and the existing ones, resulting in the addition of message transmission processes, ΔF, forming the new set of transmission processes, F_1.
  From the point of view of set theory:

$$\begin{aligned} \Delta Sch &= Sch_1 - Sch_0 \\ &= (E_1 - E_0) \cup (P_1 - P_0) \cup (F_1 - F_0) \\ &= \Delta E \cup \Delta P \cup \Delta F \end{aligned} \tag{2}$$

- The E remains unchanged, while the P and F are updated. After the initial version, $SerSch_0$, of the security scheme is verified, it evolves in an environment where the number of network entities remains unchanged, but there are changes in the logical paths. In the next version of the security scheme, $SerSch_1$, updates are made based on P_0 to form a new set of logical paths. Due to the addition of a new logical path, a new message transmission process ΔF is correspondingly added to form a new set of message transmission processes.
  From the point of view of set theory:

$$\begin{aligned} \Delta Sch &= Sch_1 - Sch_0 \\ &= (E_1 - E_0) \cup (P_1 - P_0) \cup (F_1 - F_0) \\ &= \Delta P \cup \Delta F \end{aligned} \tag{3}$$

- Other configuration updates that do not affect the E, P and F. For example, modifying the manufacturer information and configuring certain functions that do not affect the E, P and F, do not require verification.

3.3. Verification Process

The verification method for evolving a security scheme is shown in Figure 1, the number in this figure is the step of the verification process, 1 is the first step of the process. The verification process for the verification method is as follows:

Step 1: Extract the security property from the evolving security scheme, and formally express the security scheme and security property as $Sch_i = \{E, M, P, F\}$ and Φ_i.

Step 2: Use the formal tool ProVerif to verify the security scheme, which is formally expressed as $Sch \vDash \Phi$.

Step 3: Obtain the verification results of the above steps. The verification results can be classified into two situations. In one situation, ProVerif outputs counterexamples that indicate the security property is violated. In the other situation, no counterexamples are output, indicating that there is no behavior that violates the security property in the security scheme.

Step 4: Based on the verification results obtained in the previous step, follow different verification processes depending on the results. As shown inFigure 1, there are three possible situations:

- If the verification result shows that there is a counterexample, then analyze how the violated security property derives a counterexample from the model and constructs an attack path. Based on the counterexamples, improve the security scheme to address the security vulnerabilities. Additionally, since the security scheme evolution involves changes in the network environment or requirements denoted as 4*d*, the new version of the security scheme, $SerSch_{i+1}$, is a combination of improvements based on 4*a* and 4*d*.
- If the verification result shows no counterexamples and there are updated network environments or requirements, a new version of the security scheme, $SerSch_{i+1}$, is formed.
- If the verification result shows no counterexamples and no network environment or requirements are updated, the final version of the security scheme is obtained.

Step 5: Extract the formal expressions. According to the new version of the security scheme, denoted $SerSch_{i+1}$, extract the formal expression Sch_{i+1}, the incremental expression ΔSch, and the security property Φ_{i+1}.

Step 6: Verify again the updated version of Sch_{i+1} and Φ_{i+1}. Recognizing the update characteristics of the evolving security scheme, and considering the complexity of real network deployment, this paper adopts an incremental verification method to verify the security property of the security scheme.

Step 7: Finally, repeat the verification process from the above steps 3 to 6 until the verification result shows no counterexamples and no network environment or requirement updates. The verified version of the security scheme is the final version that meets the security property.

In the verification method, the network environment changes and the requirement changes of the evolving security scheme are fully simulated, and the security property of the evolving security scheme can be effectively verified.

4. Case Study

The security scheme of the IoV will evolve with the deployment scale, network nodes, and security requirements. This paper takes the evolving security scheme of the IoV as an example and applies it to the security verification framework of the evolving security scheme. According to the verification method proposed in Section 3, a formal model is constructed.

4.1. Model

4.1.1. Security Scheme of IoV

During the deployment period, the evolving security scheme of the IoV will follow the principles of increasing the number of network nodes and increasing the network size from small to large. Therefore, as shown in Figure 2, the network scale of the initial security scheme in IoV is relatively simple, with only four entities: vehicle A (*Va*), the base station (*BS*), the *Server*, and the registration authority (*RA*). The *Server* is an entity that provides resources for the IoV devices and handles some complex operations. The *BS* acts as a "middleman" responsible for forwarding messages between the vehicle and the service network. The *RA* is a system responsible for registering vehicles and managing the basic information about vehicles.

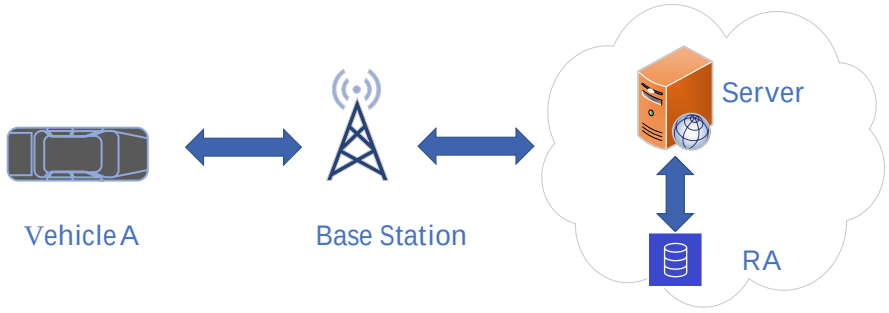

Figure 2. Communication scenario of the initial IoV.

In this communication between *Va* and the *Server*, as shown in Figure 3, the definitions of the variables in Figure 3 are shown in Table 1. The vehicle first initiates a registration application. Only after completing the registration and logging into the account, can the vehicle access subsequent services. The specific registration process is as follows: the vehicle initiates a registration request to the *RA*, and the *RA* enquires whether the vehicle information has been registered. If not, the vehicle information is registered.

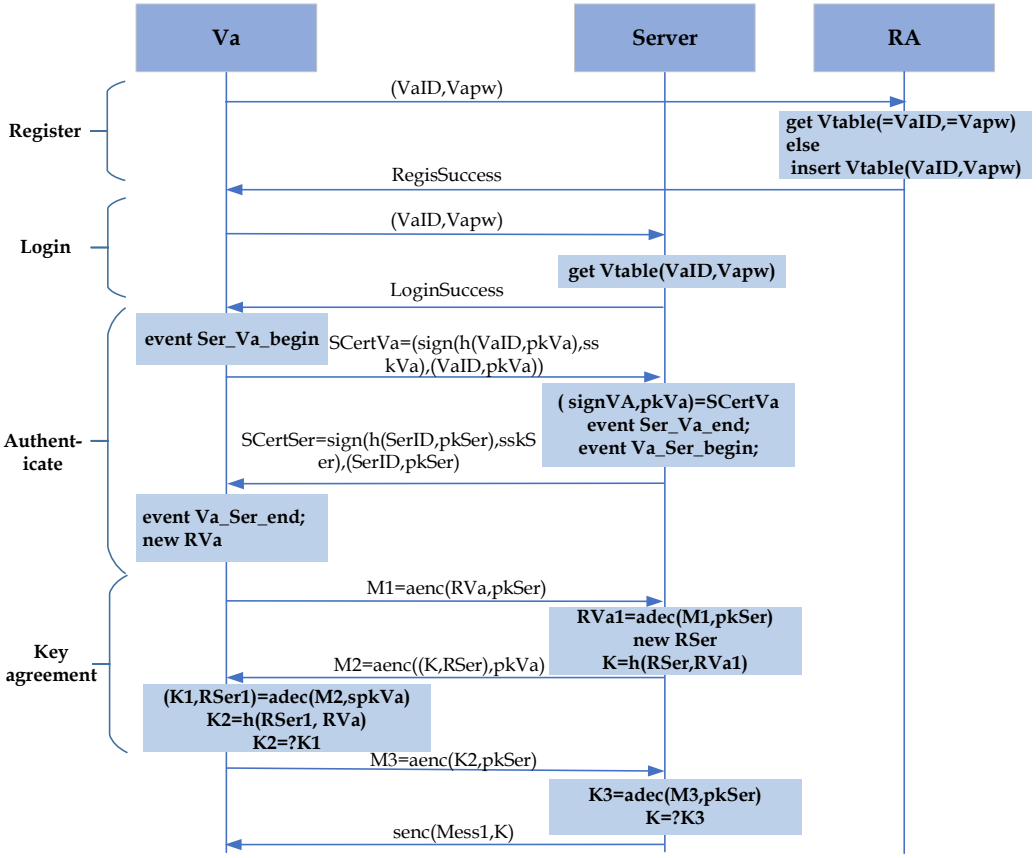

Figure 3. Communication between vehicle and server.

The communication between *Va* and the *Server* can be divided into three stages, which are: login, authentication, and key agreement. In the login phase, *Va* initiates a login information request to the *Server*. The *Server*, through *RA*, enquires whether the vehicle is a registered vehicle, and checks whether the registration information is consistent with the login request sent by *Va*. After successful login, the *Server* returns to *Va* the label *LoginSuccess* of the successful login.

Table 1. The definitions of variables in Figure 3

Variable Name	The Definition of Variable
VaID	identity document of Vehicle A
Vapw	password of Vehicle A
Vtable	vehicle table
Ser_Va_begin	beginning of server authentication to Va
Ser_Va_end	ending of server authentication to Va
Va_Ser_begin	beginning of Va authentication to server
Va_Ser_end	ending of Va authentication to server
pkVa	public key of Vehicle A
sskVa	signature secret key of Vehicle A
spkVa	signature public key of Vehicle A
SCertSer	self-signed certificate server
pkSer	public key server
sskSer	signature secret key server
SCretVa	self-signed certificate Vehicle A
RVa	random number generated by Vehicle A
RVa1	random number decrypted by server
RSer	random number generated by server
RSer1	random number decrypted by Vehicle A
M1, M2, M3, Mess1	messages between server and Va
K1, K2, K3, K4	session keys between server and Va

In the authentication phase, *Va* and the *Server* authenticate each other and obtain each other's public keys. First, *Va* initiates an authentication request to the *Server* by sending a self-signed certificate, *SCertVa*. The self-signed certificate is composed of *Va*'s basic identity information and signature. Since *Va* uses a self-signed certificate, the *Server* cannot verify its real identity. It directly considers that the authentication of *Va* has been completed, and then obtains the *pkVa* carried in *SCertVa* for subsequent communication. After obtaining *pkVa*, the *Server* also sends a self-signed certificate *SCertSer* to *Va* for authentication. Just as the *Server* cannot verify *Va*, *Va* also cannot verify the identity of the *Server*, and uses the public key *pkSer* carried in the *SCertSer* for subsequent communication.

In the key agreement phase, *Va* and the *Server* mutually agree the session key. *Va* and the *Server* generate the random numbers *RVa* and *RSer*, respectively, and then hash *RVa* and *RSer* together as the session keys. Both parties check whether the obtained session keys are consistent. If yes, the *K* is successfully agreed.

After completing the key agreement with the *Server*, *Va* uses the *K* to encrypt *Mess*1 for communication.

4.1.2. Initial Formal Model

According to the verification method proposed in Section 3, this paper constructs the tuple expression of the initial version of the security scheme in the IoV as $SchIoV_1 = \{E_1, M_1, P_1, F_1\}$. The following describes in detail the composition of each tuple in $SchIoV_1$ and the description implementation using the typed pi calculus.

- Entities. As shown in Figure 2, in the initial model, there are four entities: vehicle, *BS*, *Server*, and *RA*. Therefore, the set of entities can be represented as $E_1 = \{Va, BS, Server, RA\}$.
- Message structure. In the communication scenario of the IoV, the message structure can be divided into four types: plain text, certificate, asymmetric encryption, and symmetric encryption. The set of the message structure can be defined as $M = \{ms_1, ms_2, ms_3, ms_4\}$, where $ms_1 = Mess$. *Mess* represents the vehicle unique identifier, and labels indicating registration or login success; $ms_2 = Cert$, where *Cert* represents the structure of the self-signed certificate and the certificate issued by *CA*; $ms_3 = senc(Me, pkey)$ means the structure of the message using the *pkey* to encrypt

the Me; $ms_4 = denc(ms_3, skey)$ means the structure of the message using the $skey$ to decrypt the ms_3.

- Set of logical paths. As shown in Figure 4, the set $P_1 = \{p_{vatos}, p_{vatora}, p_{stora}\}$ can be obtained from the communication between the vehicle and the server. The p_{vatos} represents the logical path for mutual communication between Va and the $Server$, and the path specifically includes Va_BS and S_BS. The p_{vatos} indicates the logical path for mutual communication between Va and RA, and the path specifically includes Va_BS and RA_BS. The p_{stora} represents the logical path for communication between the $Server$ and RA.
- Message transmission process. From the communication process in Figure 3, the message transmission process F_1 can be obtained; that is, $F_1 = (f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}, f_{17}, f_{18})$. $f_{11} = send(Va, Server, m_{11})$ represents Va, the transmission process of sending a login message to the $Server$. $f_{12} = send(Server, Va, m_{12})$ means that the $Server$ replies to the message transmission process of the successful login of Va. $f_{13} = send(Va, Server, m_{13})$ means that Va sends a self-signature certificate. $f_{14} = send(Server, Va, m_{14})$ means that the $Server$ sends a self-signature certificate to Va. $f_{15} = send(Va, Server, m_{16})$ means that Va sends an asymmetric encrypted message to the $Server$. $f_{16} = send(Server, Va, m_{17})$ means that the $Server$ sends an asymmetric encrypted message to Va. $f_{17} = send(Server, Va, m_{18})$ means that the $Server$ sends a symmetric encrypted message to Va.

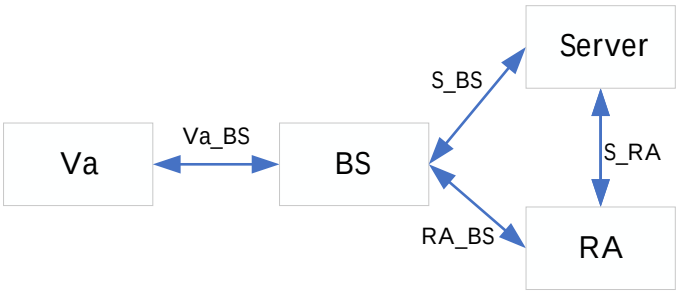

Figure 4. Communication channel.

4.1.3. Security Property

After completing the model construction of the security scheme of the IoV, it is necessary to formally define the corresponding security property. The initial version of the IoV scenario is relatively simple. Here, only two security properties are simply set up:

- Identity authentication: the $Server$ and Va can mutually authenticate each other's identities.
- Confidentiality: the confidentiality of the message $Mess1$ transmitted between the $Server$ and Va through key agreement.

Based on the above, we can define the security properties formally as $\Phi = \{Sec_0, Sec_1\}$, where Sec_0 is the identity authentication and Sec_1 is the confidentiality.

The formal expression of the security property :

$$\text{query event(Va\_Ser\_end)} ==> \text{event(Va\_Ser\_begin)} \tag{4}$$

$$\text{query event(Ser\_Va\_end)} ==> \text{event(Ser\_Va\_begin)} \tag{5}$$

These formal expressions mean that inquiring whether the authentication of Va by the $Server$ and the authentication of $Server$ by Va satisfy the relationship of the predefined sequence of events. In this context, the $event(Ser\_Va\_begin)$ indicates that the authentication of the $Server$ by Va begins; the $event(Ser\_Va\_end)$ indicates the authentication of the $Server$ by Va ends. Similarly, the $event(Va\_Ser\_\ begin)$ and $event(Va\_Ser\_$ end), respectively, indicate the beginning and end of the authentication of Va by the $Server$.

The formal expression of confidentiality:

$$\text{query attacker}(\text{Mess1}) \tag{6}$$

this formal expression queries whether the attacker can obtain the $Mess1$ between the *Server* and Va.

4.2. *Verification*

According to the formal model of the security scheme of the IoV, the formal tool ProVerif is used to verify, and then the network configuration is improved according to the verification results. At the same time, according to the requirements of deployment evolution, the security properties of the security scheme are updated, and then the updated network configuration is formally modeled and the security properties are formally expressed again, and the new formal model is verified.

4.2.1. First Round of Verification

According to the $SchIoV_1 = \{E_1, M_1, P_1, F_1\}$, and $\Phi = \{Sec_0, Sec_1\}$ constructed in Section 4.1, the formal tool ProVerif is used to verify.

Since Va uses a self-signed certificate, the *Server* cannot verify its exact identity when authenticating Va. This allows an attacker to impersonate Va and communicate with the *Server*, leading to the $event(Ser\_Va\_end)$ execute instead of $event(Ser\_Va\_begin)$. The counterexamples are shown in Figure 5.

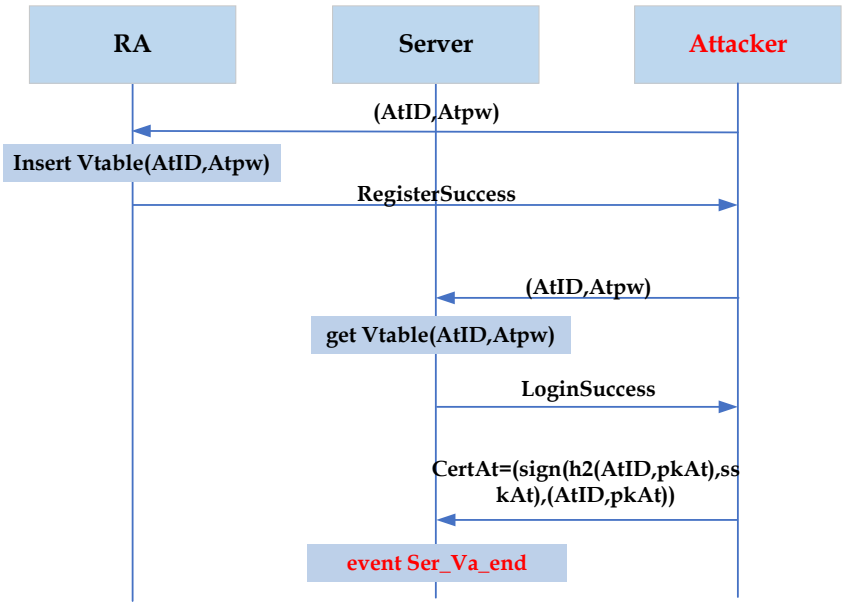

Figure 5. Counterexamples of authentication of Va by *Server*.

4.2.2. Second Round of Verification

Based on the results of the last round of verification, security flaws in Va and the *Server* authentication were discovered. This round of the authentication scheme was modified to avoid this security issue. As shown in Figure 6, the configuration of the IoV is updated to include a certificate authority (CA) , and the certificate of the *Server* is initially set to a digital certificate issued by the CA. Va applies for a digital certificate from the CA, and uses it to achieve mutual authentication with the *Server*.

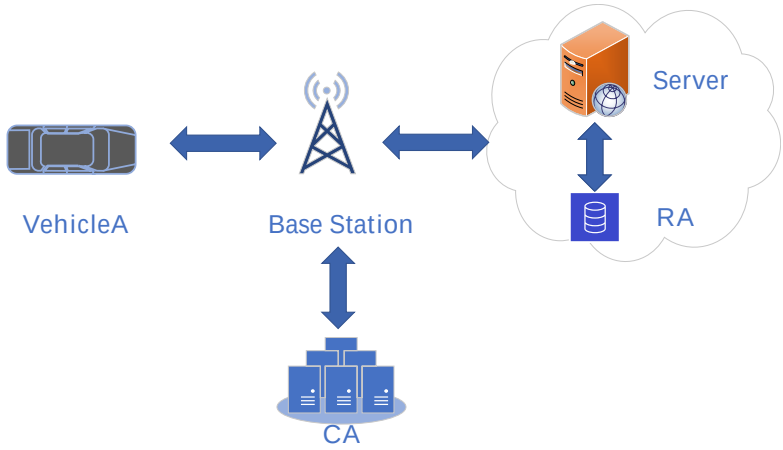

Figure 6. Communication scenario for the second round of the IoV.

As shown in Figure 7, *Va* requests the CA to issue a digital certificate, and the request message contains basic identity information, such as the unique vehicle identifier $IDVa$ and the public key $pkVa$. After verifying *Va*'s identity information, the CA first performs a hash operation on *Va*'s basic identity information to obtain a digest of *Va*'s identity information. Then, CA uses its signature private key $sskCA$ to sign the digest, and, finally, packages the signature and identity information into a digital certificate, $CerVa$. The CA issues the $CerVa$ to *Va*, and *Va* uses $CerVa$ to authenticate with the *Server*.

The specific authentication process of the *Server* for *Va* can be divided into three steps. First, after receiving the $CerVa$ sent by *Va*, the *Server* uses the signature public key of the CA to decrypt the $CerVa$ to obtain a digest of *Va*'s identity information. Then, it performs a hash operation on the $IDVa$ and $pkVa$ to obtain another digest of *Va*'s identity information. Finally, the *Server* compares whether the two digests are consistent. If the comparison results are consistent, the *Server* successfully authenticates *Va* and recognizes that $pkVa$ is *Va*'s real public key.

Similarly, the authentication process of *Va* to the *Server* is consistent with the above.

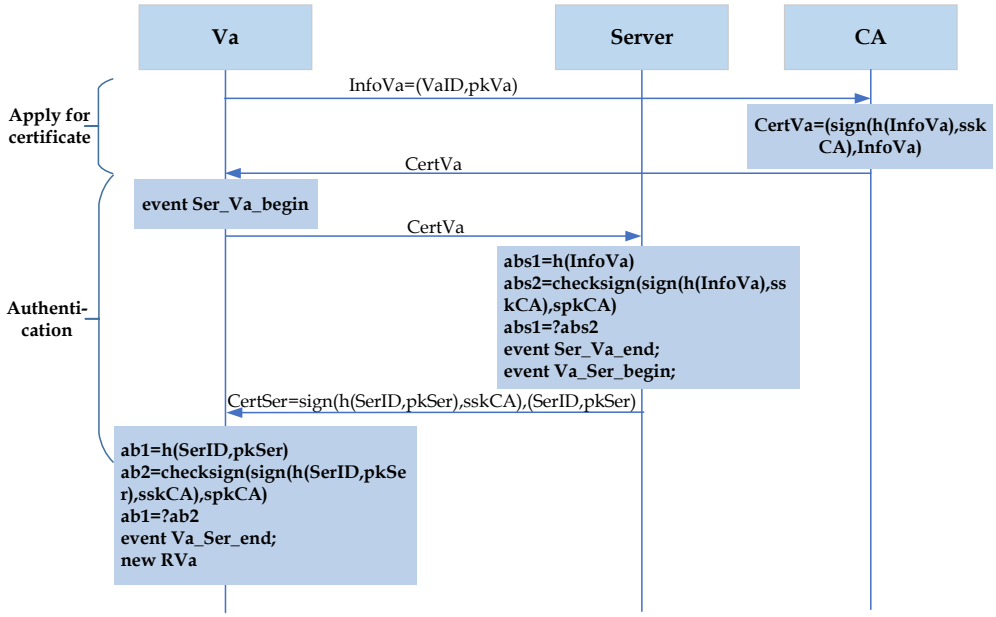

Figure 7. Mutual authentication between *Va* and the *Server*.

After modifying the security scheme, a new model is established. The model is formalized as $SchIoV_2 = \{E_2, M_2, P_2, F_2\}$. The message structure has not changed; that is,

$M_2 = M_1$. The following is a detailed introduction concerning the composition of each tuple in the formal expression:

- Entities. As shown in Figure 6, compared with the network configuration in the first round, CA is added in this round. Therefore, the set of entities is $E_2 = \{Va, BS, Server, RA, CA\}$.
- Set of logical paths. As shown in Figure 8, the set of logical paths is expanded to include the communication between Va and the CA. Hence, the set of logical paths is $P_2 = \{p_{vatos}, p_{vatora}, p_{stora}, p_{vatoca}\}$.

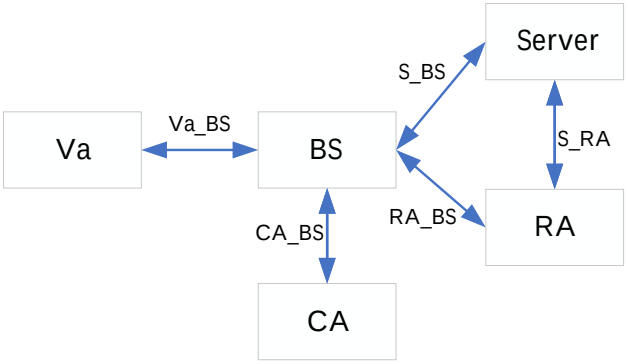

Figure 8. Communication channel for the second round of the IoV.

- Message transmission process. Compared with the first round, the message transmission process increases the communication process between Va and the CA. Therefore, the increment in the message transmission process is $\Delta F = (f_{21}, f_{22})$, where $f_{21} = (Va, CA, m_{21})$, $f_{22} = (CA, Va, m_{21})$.

This round does not change the security property, which is consistent with the first round of verification.

The verification results of this round are shown in Table 2. Va applies for a certificate from the CA, and uses the certificate to authenticate with the $Server$, enabling effective mutual identity verification.

Table 2. Verification results of the second round of the security scheme in IoV.

Security Properties	Verification Results
authentication	true
confidentiality	true

4.2.3. Third Round of Verification

In order to effectively evaluate the communication from vehicle-to-vehicle (V2V), a new vehicle entity, Vehicle B(Vb), is added in this round. In this version of the security scheme in IoV, there is no direct authentication scheme between Va and Vb. Instead, they transmit messages using public key encryption. Specifically, the Vb uses the $pkVa$ to encrypt the message $Mess2$ and sends it to Va.

The following details $SchIoV_3 = \{E_3, M_3, P_3, F_3\}$:

- Entities. As shown in Figure 9, the Vb is added in this round. Therefore, the set is $E_3 = \{Va, Vb, BS, Server, RA, CA\}$.
- Set of logical paths. As shown in Figure 10, the set of logical paths is $P_3 = \{p_{vatos}, p_{vatora}, p_{stora}, p_{vatoca}, p_{vatovb}\}$.
- Message transmission process. Compared with the second round of the message transmission process, this round increases the communication process between Va and Vb, namely, $\Delta F = (f_{31}, f_{32})$.

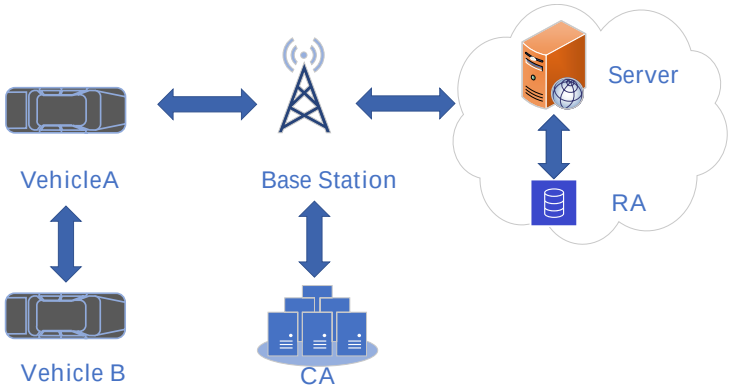

Figure 9. Communication scenario for the third round of the IoV.

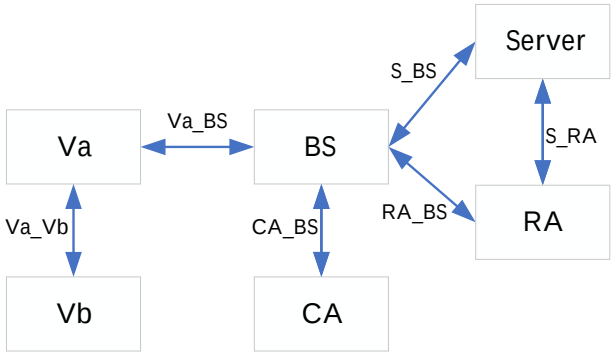

Figure 10. Communication channel for the third round of the IoV.

With the addition of Vb and the corresponding communication channel between Va and Vb, a new security property needs to be introduced:

Confidentiality: the confidentiality of the $Mess2$ transmitted by Va and Vb. Formal expression of confidentiality :

$$query \quad attacker(Mess2). \tag{7}$$

This formal expression means to query whether the attacker can obtain the $Mess2$ of the communication between Va and Vb .

In this round of the evolution process, the addition of Vb does not affect the authentication between Va and the $Server$, nor the the confidentiality of their communication. Therefore, there is no need to repeat the verification of these two properties. Only the confidentiality of the $Mess2$ of the communication between Vb and Va is incrementally verified.

Since there is no authentication scheme between Va and Vb, the communication is encrypted using the public key disclosed to each other. As the attacker can initiate a man-in-the-middle attack, $Mess2$ does not have confidentiality.

4.2.4. Fourth Round of Verification

In order to solve the confidentiality of V2V communication, this paper implements the authentication scheme based on the V2V authentication scheme proposed in [17]. As shown in Figure 11, a communication channel between Vb and the BS is added, allowing Vb to communicate with the $server$ through the BS.

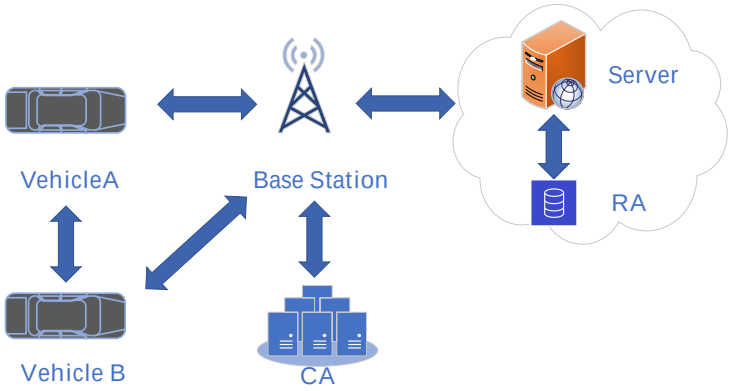

Figure 11. Communication scenario for the fourth round of the IoV.

In the security scheme of this round of the IoV, after the *Server* completes the key agreement with *Va*, the *Server* issues a symmetric key *Ka* specific for V2V communication to *Va*. *Vb* communicates with *Va* by asking the *Server* for the *Ka*. First, *Vb* logs into the *Server*, and the login steps are consistent with those in Figure 3. After the login is successful, *Vb* and the *Server* authenticate each other through the *CA* certificate, and the authentication process is consistent with that in Figure 12. *Vb* queries the *Server* for the communication key of *Va* when the mutual authentication is successful. The *Ka* is encrypted using *pkVb* and sent to *Vb* after the server checks the identity of *Vb*. *Vb* decrypts the ciphertext using its private key to obtain the *Ka*, and then uses the *Ka* to encrypt the *Mess*2 and send it to *Va*. After *Va* receives the ciphertext, it can use the *Ka* to decrypt the ciphertext to obtain *Mess*2.

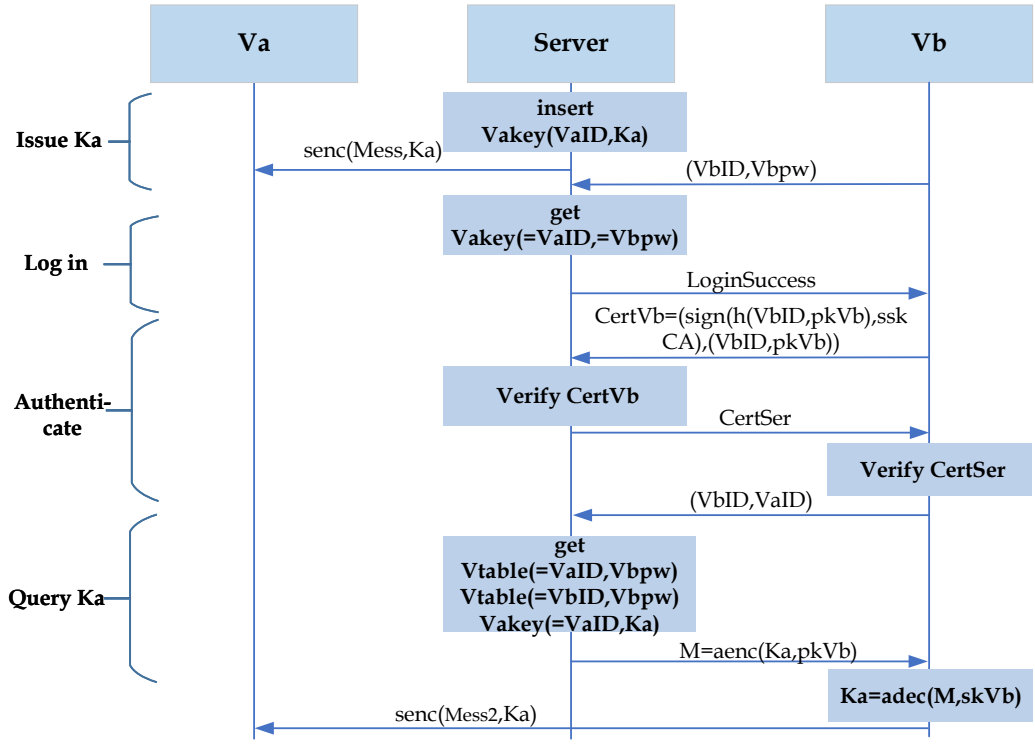

Figure 12. Communication between *Va* and *Vb*.

Compared with the security scheme of the previous version, in this round of the IoV security scheme, neither the entity nor the message structure has changed, i.e., $E_3 = E_4$, $M_3 = M_4$. The details of P_4 and F_4 are as follows:

- The set of logical paths. As shown in Figure 13, add the logic path between *Vb* and the *Server*, namely, p_{vbtos}. Therefore, $P_4 = \{p_{vatos}, p_{vatora}, p_{stora}, p_{vatoca}, p_{vatovb}, p_{vbtos}\}$.

- Message transmission process. Compared with the security scheme of the previous round of the IoV, the message transmission process increases the communication process between the *Server* and Vb, denoted $\Delta F = (f_{41}, f_{42})$.

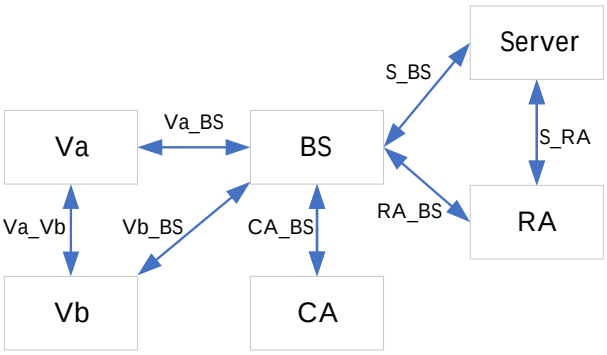

Figure 13. Communication channel for the fourth round of the IoV.

As in the previous round, the newly added channel does not affect the security properties between *Va* and the *Server*, so only *Mess*2 is incrementally verified. In the security scheme of the IoV, communication of V2V uses a dedicated key and the key is issued by the *Server*. If you want to obtain the key, you must pass the authentication of the *Server*, so you can avoid man-in-the-middle attacks, and the confidentiality of *Mess*2 is true.

5. Conclusions and Future Work

This paper proposes a verification method for an evolving security scheme, and we apply this method to the IoV scenario. First, we model the initial security scheme and the security properties in IoV, and then we use the formal method to verify the security of the security scheme. Second, according to the verification result, and the updated configuration of IoV, we push the evolution of the security scheme and validate the model of the security scheme and the security properties after updating. Finally, using a total four rounds iterate verification, we obtain a security scheme of the final version which has no security issue. The case study also shows that the method we propose is correct and usable.

In the vehicle driving environment, not everything depends on the vehicle, such as seasonal changes, for example, in the weather [32,33]; however, these may still have an impact on vehicle driving. There are relevant studies [34,35] that take such factors into consideration through LiDAR and other technologies. These factors also pose challenges to our proposed method. In the evolution of a security scheme, we need to consider relevant factors more comprehensively. Therefore, how to further improve our method and update the model represents an important direction for our future work.

Author Contributions: Conceptualization, J.Y. and C.X.; methodology, W.X.; software, C.X.; validation, W.X.; formal analysis, J.Y.; investigation, H.Z.; resources, H.Z.; data curation, W.X.; writing—original draft preparation, C.X.; writing—review and editing, W.X.; visualization, C.X.; supervision, T.J. and D.L.; project administration, J.Y. and H.Z.; funding acquisition, H.Z. and J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the China Unicom (Hainan) Innovation Research Institute Project (LThxkt202201), the Hainan Provincial Natural Science Foundation of China (620RC562), and the National Natural Science Foundation of China (62162021).

Data Availability Statement: Not applicable.

Acknowledgments: Many thanks to the editors and reviewers for their comments and help.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lu, Z.; Qu, G.; Liu, Z. A survey on recent advances in vehicular network security, trust, and privacy. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 760–776. [CrossRef]
2. Bagga, P.; Das, A.K.; Wazid, M.; Rodrigues, J.J.; Park, Y. Authentication protocols in Internet of vehicles: Taxonomy, analysis, and challenges. *IEEE Access* **2020**, *8*, 54314–54344. [CrossRef]
3. Sutrala, A.K.; Bagga, P.; Das, A.K.; Kumar, N.; Rodrigues, J.J.; Lorenz, P. On the design of conditional privacy preserving batch verification-based authentication scheme for Internet of vehicles deployment. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5535–5548. [CrossRef]
4. Yoshizawa, T.; Singelée, D.; Muehlberg, J.T.; Delbruel, S.; Taherkordi, A.; Hughes, D.; Preneel, B. A Survey of Security and Privacy Issues in V2X Communication Systems. *ACM Comput. Surv.* **2023**, *55*, 1–36. [CrossRef]
5. Fan, B.; Zhang, Y.; Chen, Y.; Meng, L. Intelligent vehicle lateral control based on radial basis function neural network sliding mode controller. *CAAI Trans. Intell. Technol.* **2022**, *7*, 455–468. [CrossRef]
6. Li, L.; Li, T. Traceability model based on improved witness mechanism. *CAAI Trans. Intell. Technol.* **2022**, *7*, 331–339. [CrossRef]
7. ONF. White Paper (Software-Defined Networking: The New Norm for Networks). Available online: https://opennetworking.org/sdn-resources/whitepapers/software-defined-networking-the-new-norm-for-networks/ (accessed on 12 October 2023).
8. Giust, F.; Sciancalepore, V.; Sabella, D.; Filippou, M.C.; Mangiante, S.; Featherstone, W.; Munaretto, D. Multi-Access Edge Computing: The Driver Behind the Wheel of 5G-Connected Cars. *IEEE Commun. Stand. Mag.* **2018**, *2*, 66–73. [CrossRef]
9. Duan, W.; Gu, J.; Wen, M.; Zhang, G.; Ji, Y.; Mumtaz, S. Emerging Technologies for 5G-IoV Networks: Applications, Trends and Opportunities. *IEEE Netw.* **2020**, *34*, 283–289. [CrossRef]
10. Zhang, J.; Letaief, K.B. Mobile Edge Intelligence and Computing for the Internet of Vehicles. *Proc. IEEE* **2020**, *108*, 246–261. [CrossRef]
11. Zhang, T. Securing Connected Vehicles: Challenges and Opportunities. 2015. Available online: http://sites.ieee.org/denver-com/files/2016/02/IoV-Security-Challenges-and-Opportunities-zhang.pdf (accessed on 23 September 2023).
12. Ghosal, A.; Conti, M. Security issues and challenges in V2X: A Survey. *Comput. Netw.* **2020**, *169*, 107093. [CrossRef]
13. Reger, L. Addressing the Security of the Connected Car. 2014. Available online: http://blog.nxp.com/addressing-the-security-of-the-connected-car/ (accessed on 23 September 2023).
14. Zhang, L.; Wu, Q.; Qin, B.; Domingo-Ferrer, J.; Liu, B. Practical Secure and Privacy-Preserving Scheme for Value-Added Applications in VANETs. *Comput. Commun.* **2015**, *71*, 50–60. [CrossRef]
15. Emara, K.; Woerndl, W.; Schlichter, J. On Evaluation of Location Privacy Preserving Schemes for VANET Safety Applications. *Comput. Commun.* **2015**, *63*, 11–23. [CrossRef]
16. Wazid, M.; Bagga, P.; Das, A.K.; Shetty, S.; Rodrigues, J.J.; Park, Y. AKM-IoV: Authenticated key management protocol in fog computing-based Internet of vehicles deployment. *IEEE Internet Things J.* **2019**, *6*, 8804–8817. [CrossRef]
17. Vasudev, H.; Deshpande, V.; Das, D.; Das, S.K. A lightweight mutual authentication protocol for V2V communication in internet of vehicles. *IEEE Trans. Veh. Technol.* **2020**, *69*, 6709–6717. [CrossRef]
18. Blanchet, B. An efficient cryptographic protocol verifier based on prolog rules. In Proceedings of the 14th IEEE Computer Security Foundations Workshop, Cape Breton, NS, Canada, 11–13 June 2001; pp. 82–96. [CrossRef]
19. Zhang, J.; Yang, L.; Cao, W.; Wang, Q. Formal analysis of 5G EAP-TLS authentication protocol using proverif. *IEEE Access* **2020**, *8*, 23674–23688. [CrossRef]
20. Basin, D.; Dreier, J.; Hirschi, L.; Radomirovic, S.; Sasse, R.; Stettler, V. A formal analysis of 5G authentication. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 1383–1396. . [CrossRef]
21. Cremers, C.; Dehnel-Wild, M. Component-based formal analysis of 5G-AKA: Channel assumptions and session confusion. In Proceedings of the 26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, CA, USA, 24–27 February 2019; The Internet Society: Reston, VA, USA, 2019.
22. Edris, E.K.K.; Aiash, M.; Loo, J.K.K. Formal verification and analysis of primary authentication based on 5G-AKA protocol. In Proceedings of the 2020 Seventh International Conference on Software Defined Systems (SDS), Paris, France, 20–23 April 2020; pp. 256–261. [CrossRef]
23. Abdullayeva, F. Internet of Things-based healthcare system on patient demographic data in Health 4.0. *CAAI Trans. Intell. Technol.* **2022**, *7*, 644–657. [CrossRef]
24. Feng, H.; Li, H.; Pan, X.; Zhao, Z.; Cactilab, T. A Formal Analysis of the FIDO UAF Protocol. In Proceedings of the NDSS, Virtual, 21–25 February 2021. [CrossRef]
25. Feng, H.; Guan, J.; Li, H.; Pan, X.; Zhao, Z. FIDO Gets Verified: A Formal Analysis of the Universal Authentication Framework Protocol. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 4291–4310. [CrossRef]
26. Bhargavan, K.; Blanchet, B.; Kobeissi, N. Verified models and reference implementations for the TLS 1.3 standard candidate. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 483–502. [CrossRef]
27. Cremers, C.; Horvat, M.; Hoyland, J.; Scott, S.; van der Merwe, T. A comprehensive symbolic analysis of TLS 1.3. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 1773–1788.

28. Saleem, M.A.; Mahmood, K.; Kumari, S. Comments on "AKM-IoV: Authenticated key management protocol in fog computing-based internet of vehicles deployment". *IEEE Internet Things J.* **2020**, *7*, 4671–4675. [CrossRef]
29. Bojjagani, S.; Reddy, Y.C.A.P.; Anuradha, T.; Rao, P.V.V.; Reddy, B.R.; Khan, M.K. Secure Authentication and Key Management Protocol for Deployment of Internet of Vehicles (IoV) Concerning Intelligent Transport Systems. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 24698–24713. [CrossRef]
30. Al-Shareeda, M.A.; Anbar, M.; Manickam, S.; Yassin, A.A. Vppcs: Vanet-based privacy-preserving communication scheme. *IEEE Access* **2020**, *8*, 150914–150928. [CrossRef]
31. Zelle, D.; Lauser, T.; Kern, D.; Krauß, C. Analyzing and securing SOME/IP automotive services with formal and practical methods. In Proceedings of the 16th International Conference on Availability, Reliability and Security, Vienna, Austria, 17–20 August 2021; pp. 1–20.
32. Onesimu, J.A.; Kadam, A.; Sagayam, K.M.; Elngar, A.A. Internet of things based intelligent accident avoidance system for adverse weather and road conditions. *J. Reliab. Intell. Environ.* **2021**, *7*, 299–313. [CrossRef]
33. Zakharov, D.; Magaril, E.; Rada, E.C. Sustainability of the Urban Transport System under Changes in Weather and Road Conditions Affecting Vehicle Operation. *Sustainability* **2018**, *10*, 2052. [CrossRef]
34. Xia, X.; Bhatt, N.P.; Khajepour, A.; Hashemi, E. Integrated Inertial-LiDAR-Based Map Matching Localization for Varying Environments. *IEEE Trans. Intell. Veh.* 2023, 1–12. *Early Access*. [CrossRef]
35. Xia, X.; Meng, Z.; Han, X.; Li, H.; Tsukiji, T.; Xu, R.; Zheng, Z.; Ma, J. An automated driving systems data acquisition and analytics platform. *Transp. Res. Part C Emerg. Technol.* **2023**, *151*, 104120. [CrossRef]