Secure Protocol For Inter-Vehicle Communication Networks

Nader Mazen Rabadi and Syed Masud Mahmud

Electrical and Computer Engineering Department, Wayne State University, Detroit, MI 48202

ABSTRACT

Security is the main concern in wireless communications. Future vehicles may use wireless Inter-Vehicle Communication (IVC) networks to exchange data among each other. IVC networks provide drivers with updated information on traffic and road conditions for efficient and safe driving. The main challenges in designing secure IVC networks are vehicle authentication and data integrity. Vehicle authentication requires IVC networks to verify that the information is sent and received by trusted vehicles. Data integrity requires IVC networks to verify that the safety-critical information will be exchanged among vehicles without illegitimate changes to their contents. In this paper, we propose a secure protocol for IVC networks. The protocol is based on the public key cryptography. Our protocol provides mutual authentication of vehicles and data integrity to IVC networks. The paper presents a detailed description of our protocol and its feasibility study in terms of processing time and latency.

1 INTRODUCTION

Recent research in the development and deployment of Intelligent Transportation Systems (ITS) are based on Inter-Vehicle Communication (IVC) networks. ITS technologies will reduce the risk of traffic accidents, improve safety and solve traffic problems [1] [2]. Future vehicles may use wireless IVC networks to exchange data among each other. IVC networks provide drivers with updated information on traffic and road conditions for efficient and safe driving. IVC networks are considered a subclass of Mobile Ad Hoc Networks (MANET). In MANET, mobile nodes exchange information among each other without any centralized infrastructure. Nodes that are within the wireless range of each other can communicate directly. Nodes that are out of the range of each other can communicate through intermediate nodes that serve as routers. To exchange data between nodes in MANET, several routing protocols were proposed, such as GPSR [3], Compass Routing [4], DREAM [5], and LAR [6].

The routing protocols in MANET involve the discovery and maintenance of possible routes between a source and a destination node. In a route discovery, a priori knowledge of the destination node is required. In general, a source node sends a route-discovery-request message to every other node in the network. As the route-discovery-request message propagates through nodes, the route discovered is added to the route-discovery-request message for the next recipient node. When the destination node receives the message, the destination node responds by sending a route-reply message to the sender node. The route-reply message propagates through the discovered route. The discovered route may include identification information or geographic locations of the visited nodes. The source node then uses the discovered route to forward its data to the destination node. Route maintenance is the mechanism by which a sending node detects if that route is broken.

Because of the use of open medium and the lack of an infrastructure, it is difficult to exchange secure messages in MANET based on untrustworthy nodes. To address these concerns, several secure routing protocols have been proposed: SRP [7], Ariadne [8], and SEAD [9]. These secure routing protocols are built on the route discovery and maintenance mechanisms. Therefore, each node in a discovered route must cooperate in identifying itself as a trusted node. Also, each node must participate in securely passing a message to the destination node and detecting any illegitimate changes to both the message and the discovered route.

IVC networks demonstrate different characteristics than MANET [10]. The difference is due to vehicles' high mobility and unpredictable drivers’ behavior. Some characteristics of IVC networks include frequent changes in network topology and fragmentation. In safety critical applications, such as collision avoidance, a priori knowledge of a destination vehicle, a route discovery and maintenance are not required. In IVC networks, a vehicle broadcasts its driving conditions to other vehicles within its wireless range. Therefore, it is unfeasible to apply MANET secure routing protocols to IVC networks. The basic challenges in designing secure IVC networks are vehicle authentication and data integrity. Vehicle authentication requires IVC networks to verify that the information is sent and received by a trusted vehicle. An intruder can masquerade as a trusted participant in IVC networks and can transmit inaccurate safety-critical information to other vehicles. This may cause catastrophic incidents and damages. Data integrity requires IVC networks to verify that the safety-critical information will be exchanged among vehicles without illegitimate changes to their contents. An intruder can
record messages being exchanged between vehicles. In an insecure network, the intruder can alter the contents of these messages and then re-transmit them back to vehicles. The new contents of these messages may as well cause false alarm of vehicles and road conditions and may cause catastrophic incidents.

In this paper, we propose a secure protocol for IVC networks. The protocol is based on the public key cryptography. Our protocol provides mutual authentication of vehicles and data integrity to IVC networks. The paper presents a detailed description of our protocol and discusses its feasibility in terms of processing time and latency.

The remainder of this paper is organized as follows. Section 2 describes the format of any message that is to be exchanged between vehicles. It also describes the steps that are required to support mutual authentications. Section 3 describes our proposed secure protocols. Section 4 describes two algorithms in which our proposed protocol can be used. Section 5 shows our performance evaluation. Finally, Section 6 discusses the conclusion of our work.

2 AUTHENTICATION STEPS AND MESSAGE FORMAT

The format for all messages to be exchanged between two vehicles is shown in Figure 1. A message consists of three parts. The first part is the data packet that contains pieces of information relevant to the IVC application, such as, road conditions, weather conditions, and vehicle conditions. It also contains information that is used in our secure protocol, such as, generated random numbers and session keys, which are explained in Section 3. The second part of the message is the certificate of the user of the vehicle and the certificate's digital signature. The user applies for this certificate at a certificate authority (CA). The certificate is the standard public-key certificate framework X.509 [14]. The general process of issuing a certificate and its digital signature is shown in Figure 2. A message digest, \( md_1 \), is generated from the certificate by applying a hash function, \( h_1 \). A hash function is a function that converts any size of an input into a fixed smaller size of output. The hash function is a one-way function that is impossible to revert. Also, the hash function should not produce the same output for two different inputs. In Figure 2, the \( md_1 \) is encrypted using the CA private key. The outcome of the encryption process is the digital signature, \( ds_1 \). The \( ds_1 \) is then included with the certificate. This process is performed only once when the user applies for a certificate at the CA. We assume that the user's certificate is embedded in a memory installed in user's vehicle. The third part of the message is another digital signature, as shown in Figure 3. The user generates a message digest \( md_2 \) from both the first and the second parts of the message using a hash function, \( h_2 \). Then, the user encrypts \( md_2 \) using the user's private key. The outcome of the encryption process is the digital signature, \( ds_2 \).

![Figure 1. Message Format](image1)

![Figure 2. Generating a Digital Signature for a Certificate](image2)

![Figure 3. Generating a Digital Signature for Data and Certificate](image3)
The purpose of generating such a message is to let the receiver of this message be able to verify the authenticity of the user and the message. The verification process consists of two steps, as shown in Figure 4. The first step is to verify the authenticity of the user. The receiver gets the certificate and applies the same hash function, \( h_2 \), to generate a message digest, \( md_2 \). Then, the receiver gets the digital certificate, \( ds_2 \), and decrypts it using the CA public key to obtain the original message digest, \( md_1 \). If both \( md_1 \) and \( md_2 \) are equal, then it indicates that the certificate is authenticated and has not been tampered with. We assume that the CA public key is also embedded in a memory installed in user’s vehicle. The receiver then uses the public key of the sender, which is contained in the certificate, to verify that the message itself is authenticated and has not been changed during transmission. As shown in Figure 5, the receiver gets the data packet and the certificate and applies the same hash function, \( h_2 \), to generate a message digest, \( md_2 \). Then, the receiver gets the digital signature, \( ds_2 \), and decrypts it using the sender’s public key to obtain the original message digest, \( md_0 \). If both \( md_0 \) and \( md_2 \) are equal, then the message is authentic.

Figure 4. Verifying the Authenticity of a Certificate

Figure 5. Verifying the Authenticity of a Message

3 PROPOSED SECURE PROTOCOL

Our proposed secure protocol is a handshake protocol that consists of either four or five steps. The number of steps depends on whether a vehicle has a session key or not. The session key will be used after the handshake protocol as an encryption/decryption key when exchanging messages. Figure 6 shows the 4-step protocol and Figure 7 shows the difference between the two protocols in Steps 4 and 5.

As shown in Figure 6, Vehicle A wants to exchange secure information in an IVC network. In Step 1, Vehicle A starts by sending a request to join a secure channel to Vehicle B. The request is a message that contains Vehicle A’s certificate and a request-access data field. This message may be recorded and replayed at a later time by an eavesdropper for an attempt to gain access to the secure channel. Therefore, Vehicle B first performs the authentication steps described in previous section to verify the authenticity of Vehicle A and the message. Then, Vehicle B generates a random number, \( b_1 \), that will be used later in Step 3 to verify the freshness of the message. We use the term freshness to indicate that any generated random number has a pre-defined lifetime. If its lifetime expires, then this random number is no longer recognized and accepted by any user. In Step 2, Vehicle B encrypts its certificate and its random number, \( b_1 \), using Vehicle A’s public key and sends the message to Vehicle A.

In Step 2, the message being sent by Vehicle B may also be recorded and replayed at a later time by an eavesdropper. Therefore, Vehicle A first decrypts the message using its private key to obtain Vehicle B’s certificate and the random number, \( b_1 \). Then, Vehicle A performs the authentication steps to verify the authenticity of Vehicle B and the message. Then Vehicle A generates a random number, \( a_1 \), that will be used later in Step 4 to verify the freshness of this message.

In Step 3, Vehicle A encrypts its certificate, the random number, \( b_1 \), and the random number, \( a_1 \), using Vehicle B’s public key. This encrypted message is sent to Vehicle B. Vehicle B decrypts the message using its private key to obtain Vehicle A’s certificate, the random number, \( b_1 \), and the random number, \( a_1 \). First, Vehicle B performs the authentication steps to verify the authenticity of Vehicle A and the message. Since Vehicle B receives back its random number, \( b_1 \), then the messages being exchanged in Steps 1, 2 and 3 are fresh and are not due to a replay attack. If the message is received after the lifetime of the random number, \( b_1 \), has expired, then the message is not fresh. As a result, the authentication process fails, and Vehicle B will ignore any subsequent messages from Vehicle A. The authentication process can be started again by sending a new access-request message.

In Step 4, if Vehicle B has a session key, then Vehicle B encrypts its certificate, the session key, and the random number, \( a_1 \), using Vehicle A’s public key. Then, Vehicle B sends this message to Vehicle A. Vehicle A decrypts the message using its private key to obtain Vehicle B’s certificate, the random number, \( a_1 \), and the session key. First, Vehicle A performs the authentication steps to verify the authenticity of Vehicle B and the message. Since Vehicle A receives back its random number, \( a_1 \), then the message being exchanged in Steps 2, 3, and 4 are fresh and are not due to a replay attack. Now, Vehicle A gets the session key and can exchange information with other vehicles using this key. If the message is received after the lifetime of the random number, \( a_1 \), has expired, then the message is not fresh. As a result, the authentication
process fails, and Vehicle A sends a new access-request message.

If Vehicle B does not have a session key, then Diffie-Hellman Algorithm is used to generate one. As shown in Figure 7, Vehicle B generates two random numbers. The first random number, \( b_2 \), is used to verify the freshness of the messages being exchanged in Steps 4 and 5. The second random number, \( y \), is used in Equation (1):

\[
Y = g^y \mod n
\]  

The values of \( g \) and \( n \) are pre-defined large prime numbers that are embedded in a memory installed inside the vehicle. In Step 4, Vehicle B encrypts its certificate, the random number, \( a_1 \), the random number, \( b_2 \), and the computed number, \( Y \), using Vehicle A’s public key. Vehicle B sends this encrypted message to Vehicle A. Vehicle A decrypts the message using its private key to obtain Vehicle B’s certificate, the random number, \( a_1 \), the random number, \( b_2 \), and the computed number, \( Y \). First, Vehicle A performs the authentication steps to verify the authenticity of Vehicle B and the message. Since Vehicle A receives back its random number, \( a_1 \), then the message being exchanged in Steps 2, 3, and 4 are fresh and are not due to a replay attack. Now, Vehicle A generates only one random number, \( x \). This random number, \( x \), is used in Equation (2):

\[
X = g^x \mod n
\]

In Step 5, Vehicle A encrypts its certificate, the random number, \( b_2 \), and the computed number, \( X \), using Vehicle B’s public key. Vehicle A sends this encrypted message to Vehicle B. Vehicle B decrypts the message using its private key to obtain Vehicle A’s certificate, the random number, \( b_2 \), and the computed number, \( X \). First, Vehicle B performs the authentication steps to verify the authenticity of Vehicle A and the message. Since Vehicle B receives back its random number, \( b_2 \), then the message being exchanged in Steps 4 and 5 are fresh and are not due to a replay attack.
Vehicle A

1. Vehicle A runs the authentication steps.
2. Vehicle A gets its random # a1 back.
3. Vehicle A generates random # x.
4. Vehicle A computes $X = g^x \mod n$.

Vehicle B

1. Vehicle B runs the authentication steps.
2. Vehicle B gets its random # b1 back.
3. Vehicle B generates random # b2 and y.
4. Vehicle B computes $Y = g^y \mod n$.

Vehicle B's certificate + a1 + b2 + Y

Encryption Algorithm

Vehicle A's private key

Vehicle B's public key

Vehicle A computes the session key $k = Y^x \mod n$

Vehicle B computes the session key $k = X^y \mod n$

Figure 7. The 5-Step Protocol

Now, both vehicles can compute the session key, $k$. Vehicle B computes, $k$, in Equation (3):

$$k = X^y \mod n \quad \text{(3)}$$

Vehicle A computes, $k$, in Equation (4):

$$k = Y^x \mod n \quad \text{(4)}$$

4 SECURE ALGORITHM

In this section, we describe two cases in which our proposed protocols can be applied.

Case 1: A vehicle without a session key joins a group of vehicles that have already established a session key with each other.

Figure 8 shows a state diagram for this case. Initially in State 1, the vehicle has no session key. In State 2, a vehicle without a session key sends an access-request message to obtain a session key. If a pre-defined time $t_1$ elapses without any response from other vehicles, the vehicle will send the access-request message again. If a response is received, then, in State 3, the 4-step protocol is performed between this vehicle and one of the vehicles that has the session key. If the 4-step protocol fails, then an access-request message will be sent again. Failures can be due to unsuccessful authentication or a predefined time $t_2$ elapses without receiving any responses from any of the four steps in the protocol. If the authentication is successful, then the vehicle will get the session key and start transmitting and receiving encrypted messages using this session key in State 4. If a vehicle with a session key does not receive messages within a pre-defined time $t_3$, then the channel is considered insecure. Therefore, this vehicle starts over from State 1.

Figure 8. Case 1: Joining A Secure Connection
Case 2: A vehicle has a session key joins a group of vehicles that have a different session key.

Case 2 is triggered when a vehicle receives a message that it cannot decrypt. We assume that each vehicle maintains a table of the session keys that are used to exchange information and the related elapsed timers. The elapsed timers are used to indicate the maximum wait time to receive an encrypted message using the corresponding session key. If the elapsed timer expires without receiving a message, the corresponding session key and the elapsed timer are dropped from the table.

5 PERFORMANCE EVALUATION

It is well known that encryption is a computational-intensive task. For IVC applications, such as, vehicle collision avoidance, the processing time is an important factor in avoiding collision. If a secure protocol is too slow in its encryption process, it will not be accepted by IVC networks. Even if it provides the highest level of security, collision may still happen. Current advancements in technology [11] [12] allow us to apply encryption algorithms in such applications. In this section, we analyze our secure protocols in terms of processing speed and latency.

In [11], researchers have developed an encryption chip that performs RSA encryption/decryption at 125 Kbps using 512 bits key. We assume that a message in our application contains 200 bytes. Therefore, one encryption/decryption task takes 12.8 ms using this encryption chip. We also assume that a hash function takes 174 Kbytes/sec [13] to generate a message digest. Therefore, one hash operation takes 1.1ms.

In our protocol, it takes one encryption task to generate a digital signature, and two decryption tasks to authenticate a message, as described in Section 2. In the 4-step protocol, digital signatures are generated four times. Also, the authentication steps are performed four times. Therefore, for the mutual authentication technique, a total of 12 encryption and decryption tasks are required. In addition, there are three encryption tasks and three decryption tasks in the 4-step protocol. Therefore, there are a total of 18 encryption and decryption tasks in our 4-step protocol. Similarly, there are 23 encryption and decryption processes in the 5-step protocol.

In similar analysis, it takes one hash operation to generate a message digest, and four hash operations to authenticate a message. In the 4-step protocol, digital signatures are generated four times. Also, the authentication steps are performed four times. Therefore, there are a total of 20 hash operations in the 4-step protocol. Similarly, there are a total of 25 hash functions in the 5-step protocol.

Table I shows the processing time and latency for the two proposed handshake protocols. We assume that messages are exchanged in a wireless bandwidth of 12 Mbps.

Consider the two vehicles shown in Figure 9. Vehicle A is traveling south at a speed of a MPH. Vehicle B is traveling west at a speed of b MPH. Both vehicles will intersect at point C. Assuming that both vehicles cannot see each other, then a collision may occur at point C. To avoid collision, both vehicles will authenticate each other using our secure protocol and exchange their driving conditions. As a result, both vehicles will respond by reducing their speed and avoid collision. We calculate the minimum distance d that is required to utilize our protocol and bring the two vehicles to stop at point C.

Table I. Performance Evaluation

| Performance Analysis Using Wireless Bandwidth = 12 Mbps Message Size = 200 Bytes |
|-------------------------|---------|--------|--------|
| Protocol                | Processing Time (ms) | Latency (ms) | Total (ms) |
| 4-step                  | 252.4   | 0.53   | 252.93 |
| 5-step                  | 321.9   | 0.66   | 322.56 |

We assume that both vehicles need to establish a session key. At locations \(x_1\) and \(y_1\), both vehicles start the 5-step protocol. At locations \(x_2\) and \(y_2\), both vehicles recognize the hazardous condition, and both start decelerating to a complete stop at point C.

We assume that the 5-step protocol takes 322 ms, and exchanging the speed values of the two vehicles takes another 128 ms. A total processing time of 500ms is assumed. After the 5-step protocol, Vehicle A travels a distance:

\[
x_1x_2 = 0.73 \cdot a \text{ feet (5)}
\]

and Vehicle B travels a distance:

\[
y_1y_2 = 0.73 \cdot b \text{ feet (6)}
\]

To stop at point C, Vehicle A travels a distance:

\[
y_2C = \frac{1.075 \cdot a^2}{s_d} \text{ feet (7)}
\]

where \(s_d\) is Vehicle A’s deceleration in feet/sec\(^2\).
Also, Vehicle B travels a distance:

\[ x_2 C = \frac{1.075 \cdot b^2}{s_b} \text{ feet} \]  

(8)

where \( s_b \) is Vehicle B’s deceleration in feet/sec\(^2\).

Therefore, the minimum distance \( d \) in feet is given by:

\[ d = \sqrt{\left(0.73 \cdot a + \frac{1.075 \cdot a^2}{s_a}\right)^2 + \left(0.73 \cdot b + \frac{1.075 \cdot b^2}{s_b}\right)^2} \]  

(9)

We assume that both vehicles have equal speed and deceleration. Figure 10 shows the minimum distance, \( d \), between the two vehicles. For a speed of 75 MPH, a minimum distance of 505 feet is required if both vehicles decelerate at 10 feet/sec\(^2\). If the deceleration is 20 feet/sec\(^2\), then the minimum distance is 932 feet. Figure 11 shows the minimum distance that both vehicles travel to a complete stop at the intersection, C. For a speed of 75 MPH, a minimum distance of 357 feet is required if both vehicles decelerate at 10 feet/sec\(^2\). If the deceleration is 20 feet/sec\(^2\), then the minimum distance is 659 feet.

![Figure 9. Collision Avoidance Using 5-step Protocol](image)

![Figure 10. Distance Between Two Vehicles](image)

![Figure 11. Distance To Intersection](image)

6 CONCLUSIONS

In this paper, we presented two secure protocols for IVC networks without the use of infrastructure. We proposed to use public key cryptography, hash functions, and digital signatures in our protocols. We explained how our proposed protocols provide mutual authentications between vehicles in terms of user’s authenticity and message integrity. Once vehicles authenticate each other, a session key is provided to exchange safety-critical information in an encrypted mode. The session key can be generated between the two authenticated vehicles using Diffie-Hellman technique. We also explained that the generation of a session key is part of our proposed mutual authentication protocols. Then, we studied the feasibility of our proposed protocols in IVC networks in terms of processing time and latency. Using a hardware encryption chip, our protocols can be processed in 323ms. We showed that if two vehicles traveling at 75 MPH toward each other at an intersection, it requires a minimum distance of 505 feet between the two vehicles and a deceleration of 10 feet/sec\(^2\) to avoid this collision. Therefore, using current technology, it is possible to build a secure communication in IVC networks, and exchange safety-critical information in a timely manner.
REFERENCES


CONTACT

Mr. Nader Mazen Rabadi
Graduate Student
Department of Electrical and Computer Engineering
Wayne State University, Detroit, MI 48202.
Phone: (313) 577-3855
Fax: (313) 577-1101
Email: nrabadi@wayne.edu

Dr. Syed Masud Mahmud
Associate Professor
Department of Electrical and Computer Engineering
Wayne State University, Detroit, MI 48202.
Phone: (313) 577-3855
Fax: (313) 577-1101
Email: smahmud@eng.wayne.edu

38