Non-Repudiation Broadcast Authentication Methods for C-V2X Communication

Takaaki KASAI[†], ***Non-Member*, *and*** **Takeshi OGAWA**[†], ***Senior-Member***

SUMMARY In this paper, we propose a real-time and less-computation authentication methods for broadcast communication in V2X. The proposed methods solve the key sharing and non-repudiation problems by using an existing broadcast authentication protocol, TESLA, by defining a new mediator function between the sender and receiver and deploying it in the Communication Base Station. The proposed methods are compared with existing technology and shown their superiority.

keywords: *Broadcast Authentication, Non-Repudiation, C-V2X, TESLA*

1. Introduction

C-V2X is a technology that realizes direct communication of V2X "Vehicle To Everything" using mobile communication system (Cellular). By Using C-V2X, information detected by vehicles, traffic lights, and various sensors on the road side can broadcast without passing through a base station or Internet, and the receiver can use this information to prevent accidents before they happen [1]. In C-V2X, the sender requests to the base station to allocation of frequencies and time slot (resource blocks), and then broadcasts using those resource blocks. In consideration of the application to hazard avoidance, it is necessary to prevent falsification of communication data by MITM(man-in-the-middle) and spoofing of the transmission source. In addition, considering that the received data will be used as evidence in the event of an accident, it is also necessary to prevent repudiation of the sent data by the sender and forging the received data by the receiver. Those security requirements are satisfied by using digital signature, but considering devices with limited communication resources such as sensors, less-computation broadcast authentication technology is necessary. TESLA is a broadcast message authentication technology that can prevent falsification of transmitting data and spoofing of the sender with low computation [2][3]. However, with TESLA the root authentication keys are to be shared between the sender and receiver before communication begins, so it is a problem to distribute the keys quickly and securely between devices such as vehicles passing each other at high speed. Also, TESLA does not have a function to prevent the sender from repudiation (non-repudiation) nor the receiver from forging. A non-repudiation broadcast authentication method has been proposed [4], in which a trusted Road Side Unit (RSU) create its digital signatures for packets received from broadcast senders and broadcasts the list to all receiver. With the method, the broadcast senders need not sign the packet but the receivers have to verify the signature of the RSU, so the receiver's computation amount are not reduced. Another research extended TESLA was proposed [5]. With the method, N of MAC (Message Authentication Code) headers using independent secret keys are added to every broadcast packet. A trusted mediator randomly discloses only one secret key to each receiver but hides from the sender which MAC header each receiver is validating. So the probability of successful repudiation by the sender is 1/n. However, there are problems such as an increase in the amount of broadcast data when the probability of non-repudiation is reduced.

In this paper, we propose an extended TESLA, a real-time and less-computation broadcast authentication technology for C-V2X. In the proposed methods, a 5G Communication Base Station (CBS) is used as a mediator, and the CBS distributes the root information of the TESLA message authentication key instead of the sender. This enables the receiver to authenticate the broadcasted packet by TESLA immediately after connecting to the broadcast area. In addition, a new non-repudiation MAC is added to the broadcast packet and the CBS inspects the MAC of all of the broadcast packet. As a result, proposed methods can prevent non-repudiation with a small amount of computation without increasing the broadcast data. We also qualitatively compare the proposed methods with existing technology and previous research.

2. TESLA: Timed Efficient Stream Loss-Tolerant Authentication

TESLA is a one-way keychain protocol using hash chains [2][3]. Fig.1 shows the keychain used in TESLA. As a preparation, the sender generates a keychain of disclosure keys K_i from a randomly generated seed K_0 using a one-way function F_1. then for each generated K_i, the sender generates a shared key $K_i{}'$ for MAC authentication using the one-way function F_2.K_i and $K_i{}'$ are used in the reverse order of the generation direction, decrementing i at regular intervals. At the beginning of the broadcast, the sender shares the end of the keychain of the disclosure key with the receivers as the root information K_{rt}. K_{rt} must be authenticated by a digital signature or other methods.

[†]The author is with Graduate School of System Design and Technology, Tokyo Denki University, Adachi, 120-8551, Japan

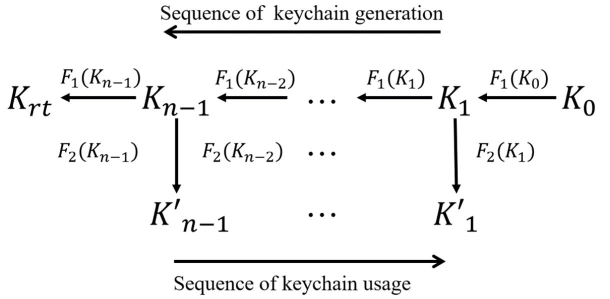

Fig. 1 Keychain generation used in TESLA

The sender then broadcasts each packet with the MAC computed using $K_i{}'$ as the key for the content of the packet. After that, the sender adds K_i corresponding to $K_i{}'$ to subsequent broadcasts packet after 2 cycles as shown in Fig. 2. To verify the disclosure key K_i, the receiver first computes K_i using F_1 and checks whether it matches the previously received disclosure key K_{i+1}. If it matches, the receiver judges that the sender is correct due to the unidirectional property of F_1. Then, $K_i{}'$ is computed using F_2, and the MAC is generated using this value as the key. If the generated MAC matches the previously received MAC information, the sender judges that the Data is justified.

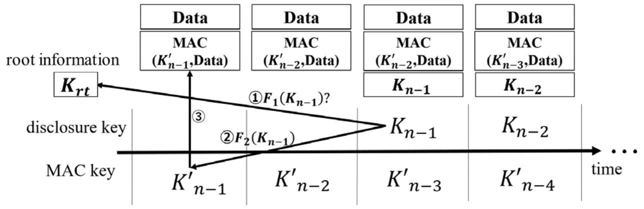

Fig. 2 Data authentication procedure using TESLA

Thus, security against MITM attack equivalent to a digital signature scheme is achieved with a low computation amount equivalent to a MAC authentication scheme for unicast. The delay time for key disclosure must be at least twice the maximum propagation delay between the sender and receiver, but since V2X is a one-hop wireless communication, the delay time for key disclosure is not expected to be a problem. However, the following two problems must be solved so as to apply TESLA to V2X communication.

(1) Sharing of keychain root information K_{rt}

With conventional TESLA, root keys K_{rt} are periodically generated and broadcasted to the receivers. If an K_{rt} for current keychain is not received, any packet in the current section cannot be authenticated. Therefore, it is difficult to apply conventional TESLA to V2X communication between devices such as vehicles passing each other at high speed.

(2) Non-repudiation

Since TESLA discloses the disclosure key to all receiver with a time lag, the sender can claim that data sent before key disclosure was generated by another receiver after key disclosure. The sender can also claim that the receiver forged the data, because the signature is based on a common secret key between the sender and the receiver. Furthermore, the receiver can use K_i, which is already disclosed, to forge data that was not actually received and claim that it was received from the sender. If the sender or the receiver claim above, the third party cannot judge the authenticity.

3. Related Work

Mujahid *et al.* [4] proposed an efficient K_{rt} distribution method in which an application server records latest K_{rt}s received from senders and distributes the K_{rt} when receivers need them. However, the non-repudiation problem was out of the scope of the proposal and didn't solve.

Farshad *et al.* [5] proposed a protocol for authenticating and non-repudiation of messages via the RSU. The sender broadcasts the data (without MAC) to the receivers and transmits the data with MAC header to the RSU via unicast. The RSU verifies the MAC, then lists the hash values of the validated packets, adds a signature, and broadcasts them to each receiver. The receiver verifies the authentication by comparing the hash value of the broadcast packet received from the sender with the hash list transmitted by the RSU. This protocol does not reduce the computation amount of the receiver because the receiver must verify the signature of the hash list received from the RSU in order to verify the broadcast packet received from the sender. In addition, since the verification of the signed hash list from the RSU also involves message authentication, the receiver must buffer the received packets within the time cycle set by the RSU, so the data in the packet cannot be used immediately.

Ayan *et al.* [6] proposed a protocol for message authentication and non-repudiation conjunction with satellite communication. The proposal extends the MAC header of broadcast packets in TESLA to n each using a different secret share key. A trusted mediator (satellite) randomly discloses only one disclosure key to each receiver via satellite link, and hides which MAC header is being monitored by the receiver. The senders broadcast data packets via internet (non-satellite broadband link, e.g., wi-fi or cellular phone). Since a MITM needs to collect all N disclosure keys to impersonate the sender, forgery (creation of N correct MAC headers) is difficult. In addition, if a sender wants to repudiation-attacks to a particular receiver, it is necessary to set correct values only for a MAC header inspected by the receiver and incorrect values for the other MAC headers. However, since the sender cannot find out the MAC header inspected by the targeted receiver, the repudiation-attack by the sender is probabilistically prevented. But, The sender can repudiation-attack successfully with probability 1/n. Also, the receiver can forge the received data and the MAC header and claim that the sender attempted a repudiation-attack. Therefore, it can be said that this protocol does not provide enough non-repudiation.

4. Proposed Scheme

Fig.3 shows the structure of the proposed scheme. TESLA is used for the broadcast from senders to receivers in a same cellular radio cell using C-V2X technologies. It is also assumed that the communication between a CBS and the senders, and the CBS and the receivers are secure using hardware authentication (Subscriber Identity Module/Authentication and Key Agreement). Also, the CBS is assumed to be universally trusted and to have sufficient computational resources.

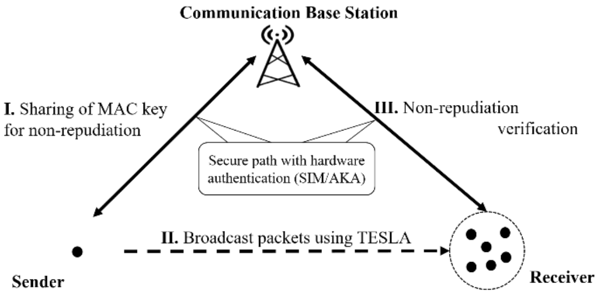
Communication Base Station
I. Sharing of MAC key for non-repudiation
III. Non-repudiation verification
Secure path with hardware authentication (SIM/AKA)
II. Broadcast packets using TESLA
Sender
Receiver

Fig. 3 Configuration of the proposed scheme

4.1. Sharing the root key K_{rt}

To solve (1) in Section 2, we propose a scheme in which the sender registers the root key K_{rt} to the CBS in advance, and the receiver receives K_{rt} from the CBS when it connects to CBS. Since secure paths are available between the CBS and the sender and between the CBS and the receiver, the CBS can guarantee the certification of K_{rt} without digital signatures.

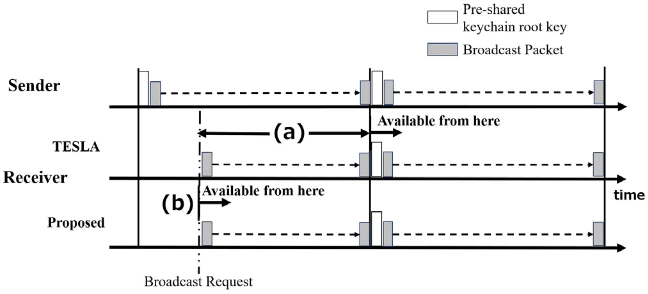

Fig. 4 Authentication delay improvement by the proposed scheme

In addition, as shown in Fig.4, with conventional TESLA, if a user joins a broadcast from the middle of the period, the data in section (a) cannot be authenticated because the K_{rt} is not shared. In the proposed methods, K_{rt} is sent from CBS to a receiver when it enters the radio cell so that the receiver can authenticate all broadcasted data after (b).

4.2. Providing non-repudiation

To solve (2) in Section 2, we propose a non-repudiation scheme by adding a procedure to TESLA and utilizing CBS that have a trust relationship with both sender and receiver.

I. Sharing MAC key for non-repudiation

In this process (Fig.3, I), the CBS generates and shares MAC key K_{S_i} for non-repudiation. Fig.5 shows the procedure.

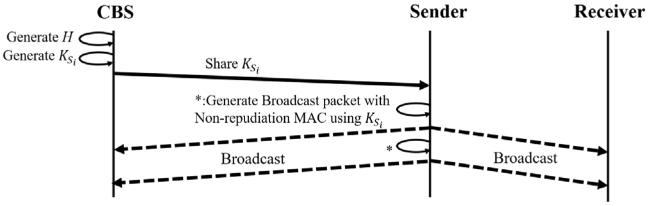

Fig. 5 MAC key K_{S_i} generation/sharing procedure

I-1. The CBS randomly generates one secret information H known only to the CBS and used by all senders.

I-2. The CBS generates a MAC key for non-repudiation. The CBS combines the unique sender ID S_iID assigned to each sender and calculate its hash.

$$K_{S_i} = h(H||S_iID)$$

I-3. The CBS shares K_{S_i} with each sender.

$$CBS \rightarrow Sender_i: K_{s_i}$$

II. Broadcast packets using TESLA

When the sender broadcasts data (Fig.3, II), generates a non-repudiation MAC using the K_{S_i}, adds it to each broadcast packet, and sends it. The structure of a packet is shown below.

$$Sender_i \rightarrow Network: \\ \{TESLApacket, MAC(TESLApacket, K_{s_i})\}$$

Here, $TESLApacket$ refers to a conventional TESLA packet and its structure is shown below.

$$\{M, TimeStamp, MAC(M, h'(K_j)), K_{j-2}, \}$$

Here, M is data to be broadcasted and includes S_iID. "$MAC(TESLApacket, K_{s_i})$" is the non-repudiation MAC.

The CBS inspects all broadcast packets, and when it detects an irregularity on non-repudiation MAC, it releases the allocated resource block for the sender and notifies the receivers of the irregularity.

III. Non-repudiation verification

If the receiver needs a certification to prevent repudiation of received data, such as when an accident occurs as a result of an action based on the received data, the receiver requests the base station to obtain the certification (Fig.3, III). Fig.6 shows the procedure.

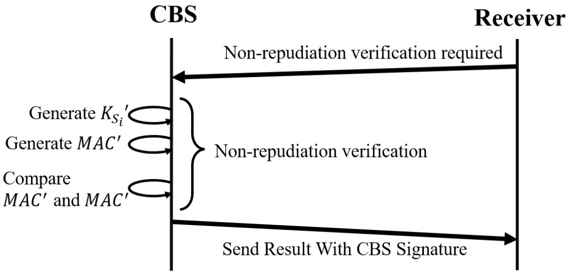
CBS
Receiver
Non-repudiation verification required
Generate K_{S_i}'
Generate MAC'
Compare MAC' and MAC'
Non-repudiation verification
Send Result With CBS Signature

Fig. 6 Non-repudiation verification procedure

III-1. The receiver forwards the packet to the base station.

$$Reciever \rightarrow CBS: \{TESLApacket', MAC\left(TESLApacket, K_{s_i}\right)'\}$$

III-2. The CBS generates K_{s_i}' from the S_iID' in the packet forwarded by the receiver and H held by the CBS.

$$K_{S_i}' = h(H||S_iID')$$

III-3. Generate MAC$'$ for the TESLA packet part. Then, compare the generated MAC$'$ with the non-repudiation MAC in the packet forwarded by the receiver.

$$MAC' = MAC(TESLApacket', Ks_i')$$

III-4. The CBS compares MAC and MAC' and the result is returned to the receiver with the signature of the CBS.
$CBS \rightarrow Reciever: (\, Result, Cert(CBS))$

Since the CBS always verifies the non-repudiation MAC in Phase II, the sender never succeeds in sending a packet that can be repudiated, except only when the receiver receives the broadcast packet correctly but the CBS fails.

Furthermore, since the receiver cannot generate a non-repudiation MAC, it cannot be claimed that the receiver has forged data that was not actually received and insists it received it from the sender.

5. Comparison and Discussion

We compare digital signatures method, previous research [4][5], and our proposed qualitatively. Table.1 shows the comparison results for each item.

As shown in Table 1, the digital signature and the previous research 1 satisfy the requirements for authentication.

Table.1 Comparison with each method
(S: sender/R: receiver)

	Digital signature	Previous research1[4]	Previous research2[5]	Proposed
Authentication of sender and prevention of message falsification	✓	✓	✓	✓
Non-repudiation of sender/ Non-forgery by receiver	✓	✓	-	✓
Sender Per-packet signature computation	RSA Signature ECDSA signature (*Computation intensive)	**S ⇒ RSU:** MAC x 1 **S⇒R:** Hash x 1	**S ⇒ R :** MAC x n	**S⇒R** MAC × 1
Mediator Per-packet verification/signature computation	Not needed	**Verification:** MAC x1 **Signature:** Hash x n Digital signature x 1	**Verification:** Hash x 1 MAC x 1 **Signature:** Digital signature x 1	**Verification:** Hash x 1 MAC x 1 **Signature:** Digital signature x 1
Reciever Per-packet verification computation	**All packets:** RSA Signature ECDSA signature	**All packets:** Digital Signature x 1 Comparison of hash	**Packets to be verified:** Digital Signature x 1	**Packets to be verified:** Digital Signature x 1

However, they are not suitable for devices with limited communication resources such as sensors because public key cryptography is required. The previous research 2 is computationally lightweight, but does not satisfy the authentication requirements. In contrast, the proposed methods satisfy all of the requirements for authentication, including non-repudiation, with a small amount of computation.

6. Conclusion

We have proposed an authentication scheme for broadcast communication that can provide non-repudiation suitable for C-V2X, using an existing protocol, TESLA. We also qualitatively compared the proposed methods with existing technology and previous research, and showed that the proposed scheme is the most suitable authentication scheme for broadcast communication in C-V2X.

References

[1] Shohei YOSHIOKA, Satoshi NAGATA "Cellular V2X Standardization in 4G and 5G", 2022

[2] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The tesla broadcast authentication protocol", 2002

[3] IETF rfc4082, "https://datatracker.ietf.org/doc/html/rfc4082",2021/10/15 cite

[4] Mujahid Muhammad, Paul Kearney, Adel Aneiba, Andreas Kunz "Efficient Distribution of Key Chain Commitments for Broadcast Authentication in V2V Communications", 2020

[5] F. R. Asl and R. Samavi, "SyNORM: Symmetric Non-Repudiated Message Authentication in Vehicular Ad Hoc Networks", 2017

[6] Roy-Chowdhury, A. Baras, J.S. "Energy-Efficient Source Authentication for Secure Group Communication with Low-Powered Smart Devices in Hybrid Wireless/Satellite Networks", 2011