Journal of Information Hiding and Multimedia Signal Processing ©2015 ISSN 2073-4212
Ubiquitous International Volume **6**, Number **2**, March **2015**

A Critique of a Lightweight Identity Authentication Protocol for Vehicular Networks

Xue-Dan Jia[1], Ya-Fen Chang[2], Chin-Chen Chang[3,4,*] and Liang-Min Wang[1]

[1]School of Computer Science and Telecommunication Engineering,
Jiangsu University, Zhenjiang, 212013, China
xdjia.ujs@gmail.com
jasonwanglm@gmail.com

[2]Department of Computer Science and Information Engineering,
National Taichung University of Science and Technology, Taichung, 40401, Taiwan
cyf@cs.ccu.edu.tw

[3]Department of Information Engineering and Computer Science,
Feng Chia University, Taichung, 40724, Taiwan

[4]Department of Computer Science and Information Engineering,
Asia University, Taichung, 41354, Taiwan
alan3c@gmail.com

Received February, 2014; revised March, 2014

Abstract. *Vehicle handover from one road-side unit to another is a common phenomenon in vehicular ad-hoc networks (VANETs). Authenticating vehicles effectively is the key to success of VANETs. Recently, Li and Liu proposed a lightweight identity authentication protocol for VANETs, which was claimed to integrate efficiency and security. However, in this paper, we show that their protocol is vulnerable to three severe drawbacks, including protocol bottleneck, location detection, and parallel session attack.*

Keywords: Vehicular ad-hoc networks, Handover authentication, Dynamic session secret process, Security analysis, Location privacy, Parallel session attack

1. Introduction.

Vehicular ad-hoc networks (VANETs) have mainly two different transmission modes, vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I), where the second mode makes plenty of entertainment services and applications provided while security is an important issue [2]. In both modes, authenticity is an essential requirement. That is, before the receiver processes a received message, the legitimacy of the sender and the integrity of the received message need to be verified to prevent illegal users from accessing VANETs services or transmitting forged messages. Because VANETs are characterized by quasi-permanent mobility [4], frequent handover authentication [3, 5, 6] is required to be performed between different road-side units and a vehicle as it is moving. Effective authentication is essential to achieve fast handover with low transmission and computation costs and security assurance.

Recently, Li and Liu proposed a lightweight identity authentication protocol (LIAP) [1] for VANETs to achieve fast handover authentication with privacy protection. To reduce handover authentication delay, dynamic session secret process (DSSP), instead of conventional cryptographic schemes, is employed to realize authentication between a vehicle and a road-side unit. They claimed that LIAP could provide efficient and secure authentication and be immune to common malicious attacks. However, through thorough analyses, we find that their protocol is neither secure nor effective as claimed. Actually, LIAP has deficiency of system performance and provides no location privacy because of the underlying pre-distribution mechanism and authentication method. Moreover, LIAP cannot resist parallel session attack such that an attacker can always impersonate a registered vehicle and be authenticated successfully by a road-side unit without knowing any authentication secrets of the vehicle.

The rest of this paper is organized as follows. Section 2 first introduces the employed authentication mechanism, DSSP, and then gives a brief review of Li and Liu's protocol, LIAP. In Section 3, the three found drawbacks of Li and Liu's protocol are shown. Finally, Section 4 concludes this paper.

2. **Review of Li and Liu's Lightweight Identity Authentication Protocol for Vehicular Networks.** Li and Liu employed a dynamic session secret process DSSP to reduce the handover authentication delay between a vehicle and a road-side unit by proposing a lightweight identity authentication protocol LIAP [1]. In this section, DSSP is first introduced, and LIAP is then reviewed.

2.1. **Review of dynamic session secret process.** Via DSSP, two entities can authenticate each other by utilizing a dynamic one-time sequence as the authentication credential. DSSP provides computational efficiency because no complicated computation operations need to be executed and only pieces of the authentication credential need to be transmitted during the authentication process. The used notations are listed as follows.
Entity1, Entity2: two entities needing to authenticate each other
X: the secret sequence shared between Entity1 and Entity2, where $X = \{x_1, x_2, \ldots, x_i\}$, and x_i represents the ith element of X
RTA: Request to Authenticate, which is a challenge generated in DSSP and in the form of a vector (r, q), where $r = \{r_i | i = 1, 2, \ldots, m$ and $1 \le r_i \le |X|\}$ and $q = \{q_i | i = 1, 2, \ldots, m$ and $1 \le q_i \le |X| - r_i + 1\}$
ATA: Answer to Authenticate, which is a response generated in DSSP and composed of a set of elements $\{(r_1, q_1) \to a_1, (r_2, q_2) \to a_2, \ldots, (r_m, q_m) \to a_m\}$, where $(r_i, q_i) \to a_i$ represents a mapping of RTA and ATA and a_i is $x_{r_i} x_{r_i+1} \ldots x_{r_i+q_i-1}$

The mutual authentication procedure of DSSP is as follows:
Step 1. Entity1 chooses and sends the first challenge RTA_1 to Entity2.
Step 2. After receiving RTA_1, Entity2 uses X, shared with Entity1, to generate ATA_1 according to RTA_1 and RTA_2. Then, Entity2 sends ATA_1 and RTA_2 to Entity1.
Step 3. After receiving ATA_1 and RTA_2, Entity1 checks whether ATA_1 is the correct answer to RTA_1 according to X. If it is not, Entity1 terminates the protocol; otherwise, Entity2 is authenticated by Entity1 successfully, and Entity1 generates ATA_2 according to RTA_2 and sends it to Entity2.
Step 4. Upon receiving ATA_2, Entity2 checks whether it is the correct answer to RTA_2. If it is not, Entity2 terminates the protocol; otherwise, Entity1 is authenticated by Entity2 successfully, and mutual authentication is achieved.

Next, we use Fig. 1 to illustrate the relationship between ATA and RTA with the shared secret X. Suppose that the shared secret X and RTA are 011010010110 and $\{(r_1, q_1)$, (r_2, q_2), (r_3, q_3), (r_4, q_4), (r_5, q_5), $(r_6, q_6)\}$= $\{(6, 2), (1, 3), (9, 4), (6, 1), (11, 2), (7, 1)\}$, respectively. The corresponding ATA is $\{a_1$, a_2, a_3, a_4, a_5, $a_6\}$ = $\{00, 011, 0110, 0, 10, 0\}$.

2.2. **Review of Li and Liu's lightweight identity authentication protocol.** Li and Liu's protocol, LIAP, for vehicular networks employs DSSP to reduce the handover authentication delay between a vehicle and a road-side unit. We first list notations used in LIAP as follows.
V_i: the ith vehicle
OBU_i: V_i's on-board unit
RSU_j: the jth road-side unit
AS: AAA server for authentication, authorization and accounting
UID_i: the identity of the user who applies for the service with V_i
PWD_i: the password of the user who applies for the service with V_i
$metaUID_i$: the transmitted identity generated by encrypting UID_i
RID_j: RSU_j's identity
K_i: the common secret key shared between V_i and AS, which is derived from UID_i and PWD_i
$F()$: a function used to compute K_i
$E_{PK}(m)$: encrypting a message m with the public key PK in an asymmetric cryptosystem
AK_U/AK_R: public key/private key of AS
$RK_{U,j}/RK_{R,j}$: public key/private key of RSU_j
N_R, N_O, N_A: random numbers
$\bigoplus$: exclusive-or operator
$\|$: concatenation operator
$A_{s,i}$: the authentication sequence shared between V_i and A_S with a limited lifetime

$$RTA = \{(r_1,q_1),(r_2,q_2),(r_3,q_3),(r_4,q_4),(r_5,q_5),(r_6,q_6)\}$$
$$= \{(6,2),(1,3),(9,4),(6,1),(11,2),(7,1)\}$$

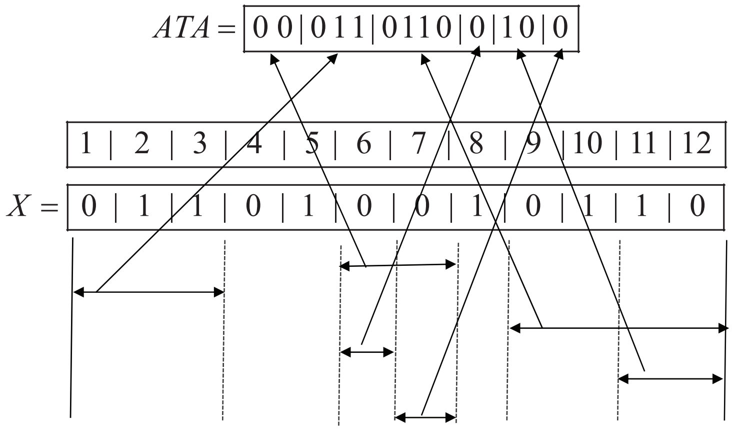
ATA = 0 0 | 0 1 1 | 0 1 1 0 | 0 | 1 0 | 0
1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12
X = 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0

Figure 1. An example of DSSP

$H()$: a hash function
S_r: a one-time session secret sequence
RTA_{Entity}: Request to Authenticate generated by *Entity* according to S_r
ATA_{Entity}: Answer to Authenticate generated by *Entity* according to S_r

In LIAP, an Internet service provider (ISP) provides V_i's user, who applies to the ISP for Internet services, with a private user account identity UID_i and an access password PWD_i. When V_i's user enters UID_i and PWD_i into OBU_i, the smart card embedded in OBU_i computes $K_i = F(UID_i \parallel PWD_i)$ and saves it. Meanwhile, AS of the ISP also computes $K_i = F(UID_i \parallel PWD_i)$ for V_i and stores it in a registration table that is maintained by AS. To protect UID_i, $metaUID_i{=}E_{AK_U}(UID_i)$ is computed in advance and stored in the smart card. LIAP is composed of three phases: initial phase, fast handover authentication phase, and renewal phase. The details are as follows.

A. Initial phase

Initial phase is triggered whenever a new vehicle joins the network, and the details are as follows:

Step 1.V_i sends $metaUID_i$ to the nearest road-side unit as a join request when it first demands a service from the VANET.

Step 2. Upon receiving $metaUID_i$ from V_i, RSU_j randomly chooses N_R and sends N_R and its own identity RID_j to V_i.

Step 3. After getting the response, V_i chooses a random value N_O and computes the authentication sequence $A_{s,i} = H(K_i \parallel N_O \parallel N_R)$. V_i then generates a random session secret sequence S_r and computes $IS_i = Sr \bigoplus A_{s,i}$. Thereafter, V_i sends $metaUID_i \parallel IS_i \parallel N_O \parallel N_R \parallel RTA_{OBU_i}$ to RSU_j.

Step 4. After receiving $metaUID_i \parallel IS_i \parallel N_O \parallel N_R \parallel RTA_{OBU_i}$ from V_i, RSU_j stores RTA_{OBU_i} and forwards $metaUID_i \parallel IS_i \parallel N_O \parallel N_R$ to AS.

Step 5. After receiving $metaUID_i \parallel IS_i \parallel N_O \parallel N_R$, AS decrypts $metaUID_i$ with its private key to get UID_i and then employs UID_i as an index to find the corresponding common secret key K_i from the registration table. Thereafter, AS computes $A_{s,i} = H(K_i \parallel N_O \parallel N_R)$ and $S_r = IS_i \bigoplus A_{s,i}$, sets a timestamp $T_{s,i}$ to control the lifetime of S_r, and chooses a random value N_A. Then, AS encrypts S_r, $T_{s,i}$ and N_A with RSU_j's public key and sends $E_{RK_{U,j}}(S_r \parallel T_{s,i} \parallel N_A)$ to RSU_j.

Step 6. After getting $E_{RK_{U,j}}(S_r \parallel T_{s,i} \parallel N_A)$ from AS, RSU_j decrypts it to extract $S_r \parallel T_{s,i} \parallel N_A$ with $RK_{R,j}$. Then RSU_j generates ATA_{RSU_j} for the received RTA_{OBU_i}, chooses RTA_{RSU_j}, and then sends $RTA_{RSU_j} \parallel ATA_{RSU_j}$ to V_i.

Step 7. After receiving $RTA_{RSU_j} \parallel ATA_{RSU_j}$, V_i first checks whether ATA_{RSU_j} is the correct answer to RTA_{OBU_i} that it has sent. If it is not, the protocol is terminated; otherwise, V_i generates ATA_{OBU_i}, which is the answer to RTA_{RSU_j}, and sends it to RSU_j.

Step 8. After getting ATA_{OBU_i}, RSU_j terminates the protocol; otherwise, authentication among V_i,

RSU_j, and AS has been achieved. Then RSU_j computes $E_{AK_U}(RID_j \parallel N_A)$ and sends it to AS.
Step 9. After getting $E_{AK_U}(RID_j \parallel N_A)$, AS decrypts it with its own private key AK_R and updates the registration table by modifying V_i's entries including authentication sequence, present connection, and timestamp.

B. Fast handover authentication phase

After initial phase, V_i can access the Internet service via RSU_j. Because vehicles in VANETs are supposed to have a high mobility, frequent handover operations are required. That is, V_i may need to access the Internet service from RSU_{j+1} instead of RSU_j, and fast handover authentication phase is triggered. To reduce the handover authentication delay, AS pre-distributes $A_{s,i}$ of V_i to either all the road-side units close to V_i's present location or the road-side units predicted by a suitable mobility model. In LIAP, to distribute $A_{s,i}$, AS chooses a random number N_A, encrypts $A_{s,i} \parallel metaUID_i \parallel T_{s,i} \parallel N_A$ with a specific road-side unit's public key, and sends the encrypted result to the specific road-side unit. When the specific road-side unit receives the encrypted data, it uses its own private key to decrypt the received data to retrieve $A_{s,i} \parallel metaUID_i \parallel T_{s,i} \parallel N_A$ and stores $A_{s,i} \parallel metaUID_i \parallel T_{s,i} \parallel N_A$ to authenticate V_i. Fast handover authentication phase of LIAP is as follows:

Step 1.V_i sends $metaUID_i$ to a new road-side unit RSU_{j+1} as a fast handover authentication request when it is moving out the communication range of the current road-side unit RSU_j.
Step 2. Upon receiving $metaUID_i$ from V_i, RSU_{j+1} checks if a corresponding entry exists. If it is not found or $T_{s,i}$ is expired, this request is rejected and renewal phase is executed; otherwise, this phase continues and RSU_{j+1} sends its RID_{j+1} to V_i.
Step 3. When V_i gets RID_{j+1}, V_i chooses a new random session secret sequence S_r and computes $IS_i = S_r \bigoplus A_{s,i}$. Thereafter, V_i sends $IS_i \parallel RTA_{OBU_i}$ to RSU_{j+1}.
Step 4. After receiving $IS_i \parallel RTA_{OBU_i}$ from V_i, RSU_{j+1} computes $S_r = IS_i \bigoplus A_{s,i}$, generates the answer to RTA_{OBU_i}, $ATA_{RSU_{j+1}}$, chooses $RTA_{RSU_{j+1}}$, and sends $RTA_{RSU_{j+1}} \parallel ATA_{RSU_{j+1}}$ to V_i.
Step 5. After getting the reply, V_i first checks whether $ATA_{RSU_{j+1}}$ is the correct answer to RTA_{OBU_i} that it has sent. If it is not, the protocol is terminated; otherwise, V_i generates ATA_{OBU_i}, which is the correct answer to $RTA_{RSU_{j+1}}$, and sends it to RSU_{j+1}.
Step 6. After receiving ATA_{OBU_i}, RSU_{j+1} checks ATA_{OBU_i}. If it is not the correct answer to $RTA_{RSU_{j+1}}$, RSU_{j+1} terminates the protocol; otherwise, mutual authentication between V_i and RSU_{j+1} has been achieved. Then RSU_{j+1} computes $E_{AK_U}(metaUID_i \parallel RID_{j+1} \parallel N_A)$ and sends it to AS.
Step 7. After getting $E_{AK_U}(metaUID_i \parallel RID_{j+1} \parallel N_A)$, AS decrypts $E_{AK_U}(metaUID_i \parallel RID_{j+1} \parallel N_A)$ with its own private key AK_R and updates V_i's present connection to RID_{j+1} in the registration table.

C. Renewal phase

In LIAP, AS sets a timestamp to $T_{s,i}$ to control the lifetime of the authentication sequence $A_{s,i}$. If $A_{s,i}$ has expired, the corresponding data will be deleted. When V_i's authentication sequence is expired and V_i attempts to connect with RSU_{j+1} for Internet services, initial phase is executed directly. In other words, renewal phase of LIAP is identical to initial phase.

3. **Drawbacks of Li and Liu's protocol.** Li and Liu declared that their LIAP could defend against a variety of common security threats. However, we find that it is vulnerable to three drawbacks. First, AS is the bottleneck of the protocol. Second, a vehicle can be easily traced. Third, an attacker can be authenticated successfully by mounting parallel session attack. In this section, the details of these found drawbacks are shown as follows.

3.1. **Protocol bottleneck.** In fast handover authentication phase of Li and Liu's protocol, AS pre-distributes $A_{s,i}$ of V_i to all the road-side units close to V_i's current location or the possible road-side units predicted by a suitable mobility model. This approach enables fast handover authentication when V_i attempts to connect with the next road-side unit RSU_{j+1}. In fast handover authentication phase, AS chooses a random value N_A and sets timestamp $T_{s,i}$ to control the lifetime of $A_{s,i}$ stored by road-side units. Then AS computes $E_{RK_{U,j+1}}(A_{s,i} \parallel metaUID_i \parallel T_{s,i} \parallel N_A)$ with the specific road-side unit RSU_{j+1}'s public key $PK_{R,j+1}$ and sends the cipher to the dedicated RSU_{j+1}. Upon receiving $E_{RK_{U,j+1}}(A_{s,i} \parallel metaUID_i \parallel T_{s,i} \parallel N_A)$, RSU_{j+1} decrypts it with its private key and stores $A_{s,i} \parallel metaUID_i \parallel T_{s,i} \parallel N_A$. If there are β road-side units to which AS needs to pre-distribute $A_{s,i}$, AS needs to execute β public-key encryption operations with β distinct road-side units' public keys. In VANETs, AS always serves a number of different vehicles, there are also lots of road-side units, and the public-key encryption will become a big burden for the whole system. Thus, AS becomes the system bottleneck.

3.2. **Location detection.** Li and Liu claimed that their LIAP could ensure a vehicle's location privacy. That is, an attacker should be unable to trace a specific vehicle by using an eavesdropping technique. However, we find that location privacy cannot be protected in LIAP. In LIAP, $metaUID_i$ is pre-calculated and stored in the smart card. Although $metaUID_i$, instead of the real identity UID_i, is transmitted, $metaUID_i$ is constant. Thus, an attacker can easily and successfully trace a vehicle by monitoring the same $metaUID_i$.

3.3. **Parallel session attack.** LIAP employs DSSP to achieve fast handover authentication between a vehicle and a new road-side unit. Li and Liu claimed that LIAP provided both efficiency and security in fast handover authentication phase. After thoroughly analyzing LIAP, we find that LIAP is vulnerable to parallel session attack. Via this attack, an attacker can impersonate V_i and be authenticated successfully by a road-side unit to access resources with the intercepted $metaUID_i$ without knowing any authentication secrets of V_i. How an attacker mounts parallel session attack in fast handover authentication phase of LIAP is depicted in Fig. 2. The details are as follows:

Suppose that RSU_{j+1} and $RSU^{'}_{j+1}$ are two road-side units that have already received V_i's authentication sequence $A_{s,i}$ from AS, and $A_{s,i}$ is not expired.

Step 1. An attacker A sends the intercepted $metaUID_i$ to RSU_{j+1} and $RSU^{'}_{j+1}$ as a fast handover authentication request.

Step 2. Upon receiving $metaUID_i$, RSU_{j+1} (or $RSU^{'}_{j+1}$) first checks its memory to find the corresponding entry. According to the assumption, the entry does exist and RSU_{j+1} (or $RSU^{'}_{j+1}$) sends its own identity RID_{j+1} (or $RID^{'}_{j+1}$) to A as the response.

Step 3. After receiving RID_{j+1} and $RID^{'}_{j+1}$, A chooses a random sequence $\widetilde{IS}$, generates RTA_A randomly, and sends $\widetilde{IS} \parallel RTA_A$ to RSU_{j+1}. Note that A has no knowledge of V_i's authentication sequence $A_{s,i}$ and does not know the one-time session secret sequence $\widetilde{S}_r$.

Step 4. Upon receiving $\widetilde{IS} \parallel RTA_A$, RSU_{j+1} computes $\widetilde{S}r = \widetilde{IS} \bigoplus A_{s,i}$ and generates the answer $ATA_{RSU_{j+1}}$ to RTA_A according to $\widetilde{S}r$. Thereafter, RSU_{j+1} chooses a new authentication request $RTA_{RSU_{j+1}}$ and sends $RTA_{RSU_{j+1}} \parallel ATA_{RSU_{j+1}}$ to A.

Step 5. When A receives $RTA_{RSU_{j+1}} \parallel ATA_{RSU_{j+1}}$, it can neither check the correctness of $ATA_{RSU_{j+1}}$ nor generate the right answer to $RTA_{RSU_{j+1}}$ by itself. So it sends $\widetilde{IS} \parallel RTA_{RSU_{j+1}}$ to $RSU^{'}_{j+1}$. Here $\widetilde{IS}$ is identical to the one that it has sent to RSU_{j+1} in Step 3, and $RTA_{RSU_{j+1}}$ is the authentication request from RSU_{j+1}.

Step 6. After getting $\widetilde{IS} \parallel RTA_{RSU_{j+1}}$, $RSU^{'}_{j+1}$ computess $\widetilde{S}r = \widetilde{IS} \bigoplus A_{s,i}$, generates the right answer $ATA_{RSU^{'}_{j+1}}$ to $RTA_{RSU_{j+1}}$ according to $\widetilde{S}r$, and chooses $RTA_{RSU^{'}_{j+1}}$. $RSU^{'}_{j+1}$ then sends $RTA_{RSU^{'}_{j+1}} \parallel ATA_{RSU^{'}_{j+1}}$ to the attacker A.

Step 7. Upon receiving $RTA_{RSU^{'}_{j+1}} \parallel ATA_{RSU^{'}_{j+1}}$ from $RSU^{'}_{j+1}$, A extracts $ATA_{RSU^{'}_{j+1}}$ and sends it to RSU_{j+1}. A just terminates the communication with $RSU^{'}_{j+1}$.

Step 8. After receiving $ATA_{RSU^{'}_{j+1}}$, RSU_{j+1} checks if the received $ATA_{RSU^{'}_{j+1}}$ is the correct answer to $RTA_{RSU_{j+1}}$. This must always hold because $ATA_{RSU^{'}_{j+1}}$ is definitely the right answer to $RTA_{RSU_{j+1}}$ because RSU_{j+1} and $RSU^{'}_{j+1}$retrieve the same $\widetilde{S}_r$ by computing $\widetilde{S}_r = \widetilde{IS} \bigoplus A_{s,i}$. So RSU_{j+1} computes $E_{AK_U}(metaUID_i \parallel RID_{j+1} \parallel N_A)$ with AS's public key and sends it to AS.

Step 9. After receiving $E_{AK_U}(metaUID_i \parallel RID_{j+1} \parallel N_A)$, AS decrypts it with its own private key AK_R and updates V_i's present connection to RID_{j+1} in the registration table.

Via the aforementioned parallel session attack, the attacker A can always be authenticated successfully in fast handover authentication phase of LIAP without knowing any authentication secrets. Moreover, this attack results in the modification of V_i's present connection in AS's registration table. This may make the legal user unable to access services in time when needed.

4. **Conclusions.** In this paper, we review the LIAP proposed by Li and Liu and show three drawbacks. First, the authentication sequence pre-distribution mechanism of LIAP makes the authentication server the bottleneck of the whole system. Besides inefficiency, LIAP cannot provide location privacy and is vulnerable to parallel session attack. In a word, even DSSP adopted in Li and Liu's protocol is simple to reduce authentication delay, LIAP is still neither efficient nor secure, and a specific vehicle can be traced. To overcome these drawbacks and achieve essential requirements in VANETs is still an important issue to have only legal vehicles access services and ensure location privacy.

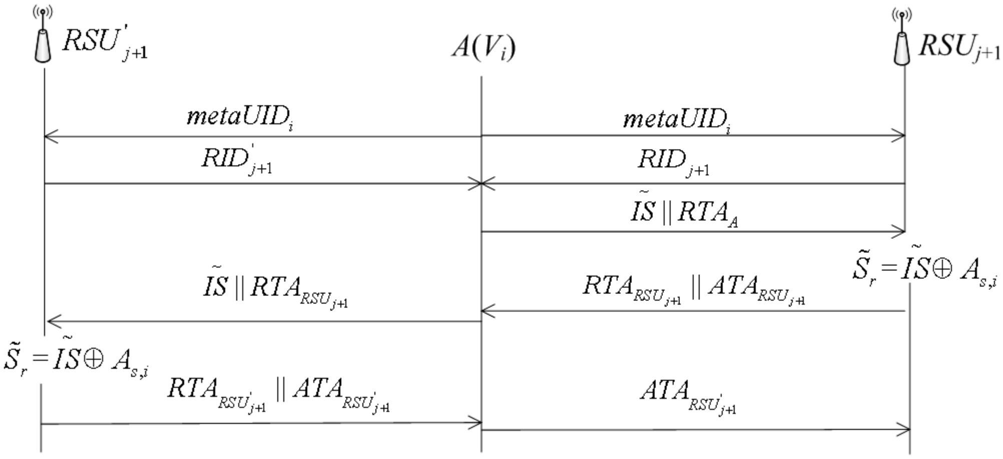
RSU'_{j+1}
$A(V_i)$
RSU_{j+1}
$metaUID_i$
$metaUID_i$
RID'_{j+1}
RID_{j+1}
$\tilde{IS} \| RTA_A$
$\tilde{S}_r = \tilde{IS} \oplus A_{s,i}$
$\tilde{IS} \| RTA_{RSU_{j+1}}$
$RTA_{RSU_{j+1}} \| ATA_{RSU_{j+1}}$
$\tilde{S}_r = \tilde{IS} \oplus A_{s,i}$
$RTA_{RSU'_{j+1}} \| ATA_{RSU'_{j+1}}$
$ATA_{RSU'_{j+1}}$

Figure 2. Parallel session attack

Acknowledgement. This research is supported by the Natural Science Foundation of China under the Grant No. 61272074, the Natural Science Foundation of Jiangsu Province under the Grant No. BK2011464. And it is also supported in part under grant number MOST 103-2221-E-025-011- from the Ministry of Science and Technology, Taiwan.

REFERENCES

[1] J.S. Li and K.H. Liu, A lightweight identity authentication protocol for vehicular networks, *Telecommunication Systems*, vol. 53, no. 4, pp. 425-438, 2013.

[2] X.D. Lin, R.X. Lu, C.X. Zhang, H.J. Zhu, P.H. Ho, and X.M. Shen, Security in vehicular ad hoc networks, *IEEE Communications Magazine*, vol. 46, no. 4, pp. 88-95, 2008.

[3] K. Mershad and H. Artail, A framework for secure and efficient data acquisition in vehicular ad hoc networks, *IEEE Trans. on Vehicular Technology*, vol. 62, no. 2, pp. 536-551, 2013.

[4] M. Raya and J.P. Hubaux, Securing vehicular ad hoc networks, *Journal of Computer Security*, vol. 15, no. 1, pp. 39-68, 2007.

[5] H. Zhu, T.T. Liu, G.H. Wei, and H. Li, PPAS: privacy protection authentication scheme for VANET, *Cluster Computing*, vol. 16, no. 4, pp. 873-886, 2013.

[6] H. Zhu, W.H. Pan, B.S. Liu, and H. Li, A lightweight anonymous authentication scheme for VANET based on bilinear pairing, *Proc. of the 4th International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, pp. 222-228, 2012.