Cryptanalysis of a Privacy-Preserving Communication Architecture for V2G Networks in Smart Grid

Qi Jiang, Jianfeng Ma, Guangsong Li and Xiang Lu

Abstract: Quite recently, Yang et al. proposed an effective security architecture to achieve secure and privacy-preserving communication and precise reward for battery vehicles (BVs) in a vehicle-to-grid (V2G) network. In this letter, we will point out their scheme has a serious security flaw, and show that the scheme fails to achieve mutual authentication between the BV and aggregators.

Keywords: Authentication, impersonation, privacy, smart grid, V2G networks

I. Introduction

In the emerging vehicle-to-grid (V2G) network, battery vehicles (BVs) communicate with the power grid to sell their electricity storage by delivering electricity into the grid as required [1]. There are 4 parties in a V2G network, including central aggregator (CAG), local aggregator (LAGs), individual BV, and trusted authority (TA). CAG communicates with the power grid directly. LAGs directly monitor all the BVs within the local area and send the collected status information to CAG in the batch mode.

It is very crucial not only to secure the communication between different parties within in V2G network against various cyber attacks, but also to protect the privacy of BVs. Yang et al.[1] proposed P^2, an effective security architecture to achieve secure and privacy-preserving communication and precise reward for BVs in a V2G network. They claim that P^2 achieves the following security goals: mutual authentication between the BV and aggregators (CAG or LAGs), confidentiality and integrity of the communication, location and identity privacy of the BV, incontestable and anonymous reward and efficient revocation of the BV.

However, in this letter, we will point out their scheme has a serious security flaw, and show that the scheme fails to achieve mutual authentication between the BV and aggregators. An adversary is able to impersonate as an eligible BV to access the LAG.

The following notations are used throughout this letter.

$\rightarrow\!\!\rightarrow$: multihop communication.

$||$: concatenation.

ID_x: the identity of an entity x in V2G network.

PS_x: the pseudonym generated by a BV x based on ID_x.

Q_x / Γ_x: the public/private key pair of an entity x corresponding to its real identity ID_x.

$Q_{\mathrm{PS}_x} / \tilde{\Gamma}_x$: the public/private key pair of a BV x corresponding to its pseudonym PS_x.

$SIG_{\Gamma_x}(M)$: the ID-based signature on a message M using signer x's private key Γ_x.

$SKE_k(M)$: the symmetric key encryption on a message M using the shared secret key k.

$HMAC_k(M)$: the hash-based message authentication code on a message M using the shared secret key k.

ST_x: the current status of a BV x.

Qi Jiang (email: jiangqixdu@gmail.com), Jianfeng Ma (e mail: jfma@mail.xidian.edu.cn), Guangsong Li (email: liguangsong@yahoo.com.cn) and Xiang Lu (email: luxiang527@gmail.com) are with Key Laboratory of Computer Networks and Information Security(Ministry of Education), Xidian University, Xi'an, China.

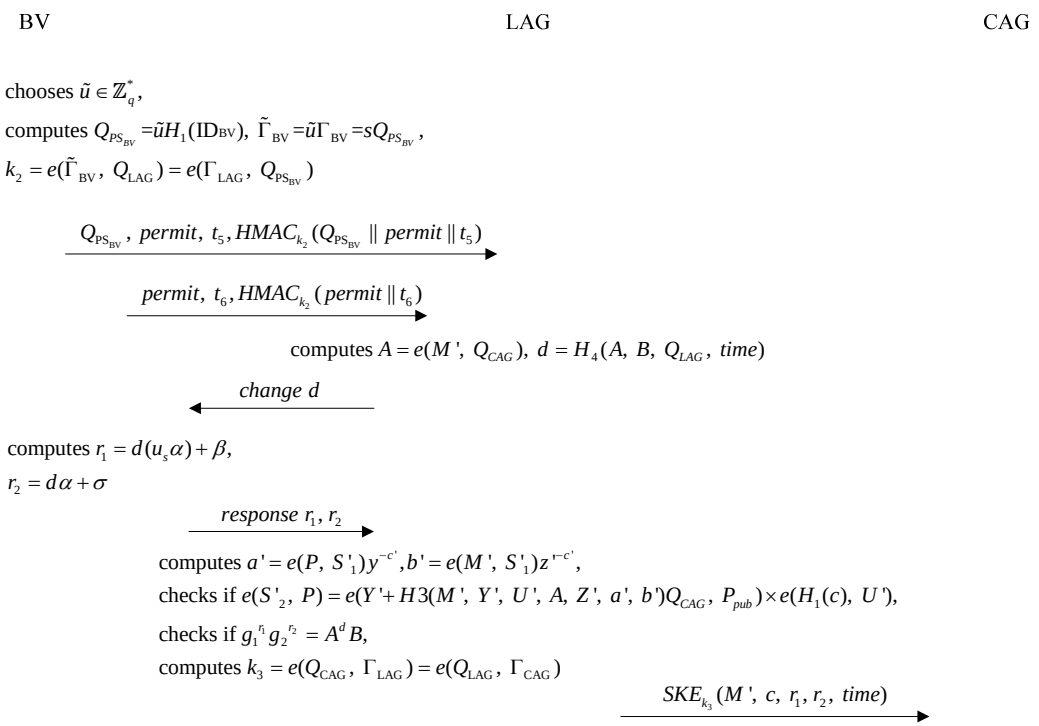


Fig. 1. *permit* verification

II. Review of P^2

P^2 consists of 4 phases: initialization, permit-based access control and BV's monitoring, anonymous service providing and rewarding, and BV's revocation protocol. In this section, we will only briefly review the first two phases, because the last two phases are not closely related to the work in this letter.

1. Initialization

TA initializes the system by performing the following algorithm.

1) Select a security parameter $k \in \mathbb{Z}^+$, and then generate a prime q, two groups G_1, G_2 of order q and an admissible bilinear map e. After that, TA chooses 3 random generators P, P_1, P_2 and 4 cryptographic hash functions: H_1, H_2, H_3, H_4.

2) Pick a random $s \in \mathbb{Z}_q^*$ as the master key, and set $P_{pub} = sP$. Then publish the system parameters params = $<q, G_1, G_2, e, P, P_{pub}, P_1, P_2, H_1, H_2, H_3, H_4>$.

Each party x, including all the BVs, LAGs, and CAG submits its identity ID_x to TA to obtain its public/private key pair Q_x / Γ_x, where $Q_x = H_1(\mathrm{ID}_x)$ and $\Gamma_x = sH_1(\mathrm{ID}_x)$.

2. *permit*-Based Access Control and BV's Monitoring

Firstly, the eligible BV registers at CAG and obtains the *permit* from CAG in advance. Then it presents this *permit* to the LAG to access the V2G network.

1) permit Generation: The ID-based restrictive partially blind signature scheme proposed in [2] is adopted for generating *permit* to ensure that the verifier can not link BV's real identity with this *permit* when the *permit* is presented later. The permit generation algorithm is omitted, please refer to [1] for the details.

2) permit Verification: To access the V2G network, each BV presents *permit*, and a pseudonym $\mathrm{PS_{BV}}$ to the LAG located in the area. The pseudonym is generated by picking a random number and computing the corresponding public/private key pair. Then, the LAG checks the validity of the *permit* and $Q_{PS_{BV}}$. If both the *permit* and $Q_{PS_{BV}}$ are valid, the LAG stores them together. For each accepted permit, the LAG records *M'*, *c*, r_1, r_2, *time* and sends them to CAG. The detail of the *permit* verification process is presented in Fig. 1.

3) BV's Monitoring: After confirming the validity of the presented *permit* and $Q_{PS_{BV}}$, a session key k'_2, which is generated based on k_2 and could be updated periodically, is used to protect the following communication between the BV and LAG. The BV reports its status ST_{BV} to the LAG when the monitoring

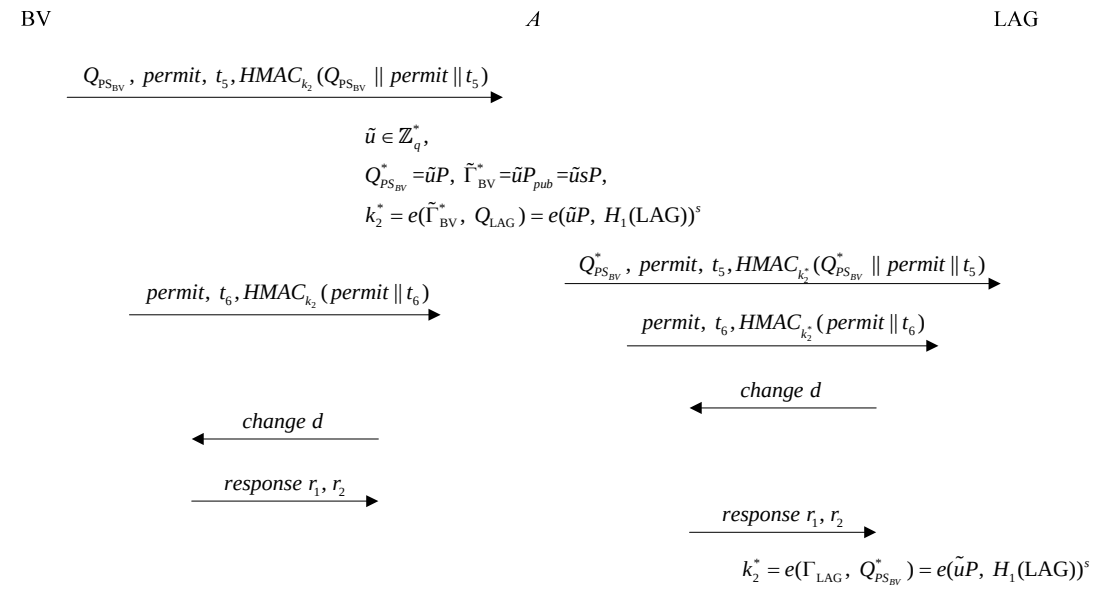


Fig. 2. Impersonation attack on P^2

cycle comes. After collecting ST_{BV} s of all the BVs in the local area, LAG forwards them to the CAG in the batch mode.

(1) BV $\rightarrow\rightarrow$ LAG: $SKE_{k'_2}(Q_{PS_{BV}}, ST_{BV}, t_7)$, $HMAC_{k'_2}(Q_{PS_{BV}} \| ST_{BV} \| t_7)$.

(2) LAG $\rightarrow\rightarrow$ CAG : $SKE_{k_3}(ST_{BV}\, s, t_8)$, $HMAC_{k_3}(ST_{BV}\, s \| t_8)$.

III. Cryptanalysis of P^2

We now show that P^2 has a serious security flaw. Specifically, an adversary *A* which is not a registered BV can forge a pseudonym and impersonate as an eligible BV to access the V2G network.

Assume that an adversary *A* obtains the public key $P_{pub} = sP$, it can cheat the LAG by the following steps, as illustrated in Fig. 2.

Step 1. After intercepting the access request message $Q_{PS_{BV}}$, *permit*, t_5, $HMAC_{k_2}(Q_{PS_{BV}} \| permit \| t_5)$ from an eligible BV, the adversary *A* first forges a pseudonym, then *A* follows the steps of the *permit* verification process.

Step 2. Upon receiving the message from *A*, the LAG follows the protocol to check if the *permit* is expired and verify its validity. In this procedure, *A* just relays the message between the BV and the LAG.

Step 3. To check the validity of the pseudonym, the LAG follows the protocol to compute $k_2^* = e(\Gamma_{LAG}, Q^*_{PS_{BV}}) = e(sH_1(ID_{LAG}), \tilde{u}P) = e(\tilde{u}P, H_1(LAG))^s$ $= e(\tilde{u}sP, H_1(ID_{LAG})) = e(\tilde{\Gamma}^*_{BV}, Q_{LAG})$.

Clearly, we know that the *HMAC* value can pass the LAG's verification.

The above attack shows that it is trivial for an adversary *A* to impersonate as an eligible BV to access the LAG. As a result, the identified design flaw inherent in P^2 makes it fail to achieve mutual authentication between the BV and LAG.

IV. Conclusion

In this letter, we pointed out an inherent design flaw in P^2. The identified design flaw allows an adversary to impersonate as an eligible BV and access the V2G network. Therefore, we recommend that P^2 should not be deployed for real world applications and hope that similar mistakes can be avoided in future design of privacy-preserving security protocols.

References

[1] Z. Yang, S. Yu, W. Lou, and C. Liu, "P^2: Privacy-Preserving Communication and Precise Reward Architecture for V2G Networks in Smart Grid," *IEEE Transactions on Smart Grid*, DOI: 10.1109/TSG.2011.2140343.

[2] X. Chen, F. Zhang, and S. Liu, "ID-based restrictive partially blind signatures and applications," *J. Syst. Softw*, vol. 80, no. 2, 2007, pp.164-171.