A Note on “Authenticated Key Agreement Protocol for Secure Communication Establishment in Vehicle-to-Grid Environment With FPGA Implementation”

Zhengjun Cao[1], Lihua Liu[2]

Abstract. We show that the key agreement scheme [IEEE Trans. Veh. Technol. 71(4): 3470-3479, 2022] fails to keep user anonymity, not as claimed.

Keywords: Key agreement, anonymity, mutual authentication, vehicle-to-grid.

1 Introduction

Recently, Sureshkumar *et al.* [1] have presented a mutual authentication and key agreement protocol in vehicle-to-grid environment. It is designed to meet many security requirements, such as mutual authentication, session key establishment, user anonymity, perfect forward secrecy, resistance to man-in-the-middle attack, off-line password guessing attack, replay attack, stolen smart card attack, insider attack, impersonation attack, and traceability attack. In this note, we remark that the scheme fails to keep user anonymity.

2 Review of the scheme

In the proposed scenario, there are different entities: Smart Electric Vehicle (SEV), Charging Station (CS), Fog server (FS), Cloud server and the Utility Service Provider (USP). The fog server controls and monitors the vehicles and charging station in the network. The USP collects data from a number of smart vehicles. Let $h : \{0,1\}^* \to \{0,1\}^n$ be a hash function, where the positive integer n is a security parameter. Let H be a bio-hash function. The USP sets its private key as s. The basic scheme can be described as follows (see Table 1).

3 The loss of user anonymity

OBSERVATION. Notice that $W_1 = L_1 \oplus L_2 \oplus C_i, W_2 = L_1 \oplus A_1 \oplus C_i$, $M_1 = \{W_1, W_2, L_2, Auth_u, T_1\}$. Since M_1 is transferred via the open channel, an adversary can capture it and retrieve W_1, W_2, L_2.

---

[1] Department of Mathematics, Shanghai University, Shanghai, 200444, China
[2] Department of Mathematics, Shanghai Maritime University, Shanghai, 201306, China.
Email: liulh@shmtu.edu.cn

Table 1: The Sureshkumar *et al.*'s key agreement scheme

User U_i: $\{U_{ID_i}, U_{PW_i}\}$	Utility Service Provider (USP): $\{s\}$	Charging Station (CS)
	$\boxed{\text{Registration}}$	
Input identity U_{ID_i}, password U_{PW_i}. Imprint the biometric *Bio* to compute $b_i = H(Bio)$, $A_1 = h(U_{ID_i}), A_2 = h(U_{PW_i} \| b_i)$. $\xRightarrow[\text{[secure channel]}]{A_1, A_2}$	Compute $S_i = h(A_1 \| s), B_i = A_2 \oplus S_i$, $C_i = h(A_2 \| B_i), D_i = B_i \oplus C_i$, $E_i = h(S_i \| C_i \| D_i)$. Build the smartcard as $SC = \langle B_i, D_i, E_i \rangle$. $\xLeftarrow{SC}$	
Compute $F_i = B_i \oplus A_2 \oplus A_1$, $K_i = h(A_1 \| b_i)$, and $G_i = A_1 \oplus (U_{PW_i} \| h_1(b_i))$, where h_1 is a hash function		Submit the identity CS_{ID_j}. $\xLeftarrow{CS_{ID_j}}$
whose output concatenated to U_{PW_i} results to the size of the output of h.	Compute $c_j = h(CS_{ID_j} \| s)$. $\xRightarrow{c_j}$	
Reconstruct the smartcard as $SC = \langle F_i, E_i, G_i, K_i \rangle$.		Keep c_j as its secret key.
	$\boxed{\text{Mutual Authentication \& Key Agreement}}$	
Enter U_{ID_i}, U_{PW_i}. Imprint *Bio*. The SC computes $b_i = H(Bio)$, $A_1 = h(U_{ID_i})$, $A_2 = h(U_{PW_i} \| b_i)$, $S_i = F_i \oplus A_1$, $B_i = A_2 \oplus S_i$, $C_i = h(A_2 \| B_i)$, $D_i = B_i \oplus C_i$. Check $E_i = h(S_i \| C_i \| D_i)$. If so, pick a nonce R_u, the timestamp T_1 to compute $L_1 = h(A_1 \| R_u)$, $L_2 = L_1 \oplus S_i$, $Auth_u = h(L_1 \| L_2 \| T_1)$, $W_1 = L_1 \oplus L_2 \oplus C_i$, $W_2 = L_1 \oplus A_1 \oplus C_i$.		Check that $\lvert T_2 - T_1 \rvert < \triangle T$. Pick a nonce R_{CS} to compute $L_3 = h(CS_{ID_j} \| R_{CS})$, $L_4 = L_3 \oplus c_j$,
	$\xrightarrow[\text{[open channel]}]{M_1=\{W_1, W_2, L_2, Auth_u, T_1\}}$	$Auth_{CS} = h(L_3 \| L_4 \| T_2)$.
	Check that $\lvert T_3 - T_2 \rvert < \triangle T$.	$\xleftarrow[W_1, W_2, L_2, Auth_u, T_1\}]{M_2=\{CS_{ID_j}, L_4, Auth_{CS}, T_2,}$
	Compute $c_j = h(CS_{ID_j} \| s)$, $L_3 = L_4 \oplus c_j$. Check if $Auth_{CS} = h(L_3 \| L_4 \| T_2)$. Then compute $A_1 = W_1 \oplus W_2 \oplus L_2$ $S_i = h(A_1 \| s)$, $L_1 = L_2 \oplus S_i$. Check that $Auth_u = h(L_1 \| L_2 \| T_1)$. Pick a nonce R_{USP} to compute $L_5 = h(T_1 \| T_2 \| T_3 \| R_{USP})$, $SK = h(L_1 \| L_3 \| L_5)$, $N_{u1} = L_3 \oplus h(L_1 \| S_i)$, $N_{u2} = L_5 \oplus h(L_1 \| S_i)$,	
	$N_{CS_1} = L_1 \oplus h(L_3 \| c_j)$, $N_{CS_2} = L_5 \oplus h(L_3 \| c_j)$, $N_{auth_{CS}} = h(N_{CS_1} \| N_{CS_2} \| c_j \| L_3)$, $N_{auth_u} = h(N_{u1} \| N_{u2} \| S_i \| L_1)$.	Retrieve N_{CS_1}, N_{CS_2} from M_3. Check $N_{auth_{CS}} = h(N_{CS_1} \| N_{CS_2} \| c_j \| L_3)$. If so, compute $L_1 = N_{CS_1} \oplus h(L_3 \| c_j)$,
Retrieve N_{u1}, N_{u2} from M_4. Check	$\xrightarrow[N_{u1},\ N_{u2},\ N_{auth_u}\}]{M_3=\{N_{CS_1}, N_{CS_2}, N_{auth_{CS}},}$	$L_5 = N_{CS_2} \oplus h(L_3 \| c_j)$.
$N_{auth_u} = h(N_{u1} \| N_{u2} \| S_i \| L_1)$.		$SK = h(L_1 \| L_3 \| L_5)$.
Compute $L_3 = N_{u1} \oplus h(L_1 \| S_i)$, $L_5 = N_{u2} \oplus h(L_1 \| S_i)$, $SK = h(L_1 \| L_3 \| L_5)$.	$\xleftarrow{M_4=\{N_{u1}, N_{u2}, N_{auth_u}\}}$	

Hence, the adversary can obtain

$$h(U_{ID_i}) = A_1 = W_1 \oplus W_2 \oplus L_2$$

The hash value is unchanged for the user in different sessions.

CLARIFICATION. The real identifier U_{ID_i} could be a regular string, and the pseudo-identifier $h(U_{ID_i})$ is a random string. In Fig.a (see Fig.1), U_{ID_i} uniquely corresponds to $h(U_{ID_i})$, and different sessions (launched by this entity) can be attributed to the unique pseudo-identifier. In this case, *the unique pseudo-identifier can be eventually used to recognize this entity.* But in Fig.b, U_{ID_i} corresponds to different pseudo-identifier $U^{(1)}_{PID_i}, \cdots, U^{(k)}_{PID_i}$. Therefore, the adversary cannot attribute different sessions to the entity, even though these sessions are launched by this entity. By the clarification, we find the scheme fails to keep user anonymity.

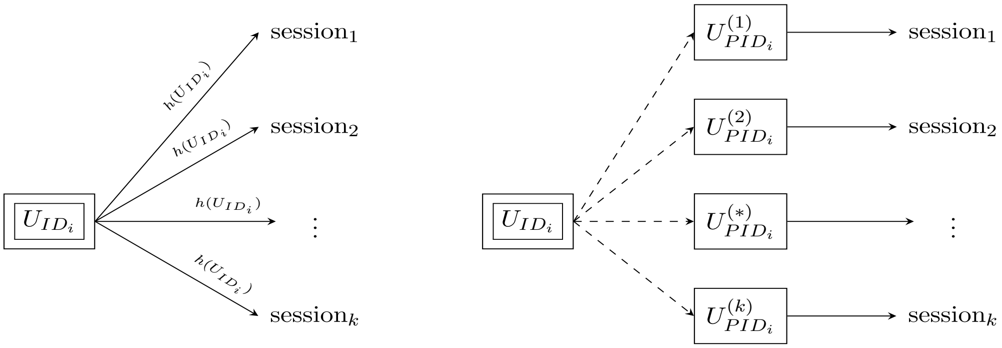

Fig.a: The false anonymity   Fig.b: The true anonymity

Figure 1: The false anonymity versus true anonymity

DISCUSSION. As we know, the identity of a person or thing is the characteristics that distinguish it from others. So, a member's identifier in the system is public and available [2]. Suppose Υ is the set of all identifiers in the system. Usually, it has a moderate size. The adversary who has captured M_1 and retrieved the pseudo-identifier $h(U_{ID_i})$, can test

$$h(U_{ID_i}) = h(\chi), \quad \chi \in \Upsilon$$

Once such an identity χ is searched out, the adversary can affirm that $\chi = U_{ID_i}$ due to the collision-free property of the hash function h. That means the user's real identity U_{ID_i} can also be recovered.

4 Conclusion

We show that the Sureshkumar *et al.*'s key agreement scheme is flawed. The scheme simply acknowledges that user anonymity is equivalent to protecting the target user's identity against exposure, while the hash value of identity can be exposed. We want to clarify that the true anonymity means that an adversary cannot attribute different sessions to different target users, even though the adversary cannot recover the true identifier from the hash value. We hope the findings in this note could be helpful for the future work on designing such key agreement schemes.

References

[1] V. Sureshkumar, P. Chinnaraj, P. Saravanan, Ruhul Amin, J. Rodrigues: Authenticated Key Agreement Protocol for Secure Communication Establishment in Vehicle-to-Grid Environment With FPGA Implementation. IEEE Trans. Veh. Technol. 71(4): 3470-3479 (2022)

[2] Z. Cao: A Note on "efficient Provably-Secure Dynamic ID-Based Authenticated Key Agreement Scheme With Enhanced Security Provision", IEEE Trans. Dependable Secur. Comput., doi: 10.1109/TDSC.2023.3302300.