A Note on "A Secure Anonymous D2D Mutual Authentication and Key Agreement Protocol for IoT"

Zhengjun Cao[1], Lihua Liu[2]

Abstract. We show that the key agreement scheme [Internet of Things, 2022(18): 100493] is flawed. (1) It neglects the structure of an elliptic curve and presents some false computations. (2) The scheme is insecure against key compromise impersonation attack.

Keywords:Device to Device communication, Key agreement, Elliptic curve, Internet of Things, Key compromise impersonation attack

1 Introduction

Recently, Hajian, Haghighat, and Erfani [1] have presented an anonymous Device-to-Device mutual authentication and key agreement protocol for IoT, in which there are two entities: IoT devices and trusted authority (TA). These devices include embedded sensors in smart vehicles, intelligent health systems, and other single-hop or hierarchical networks. They communicate with their peers or remote servers without human involvement. TA provides offline information for IoT devices. In the considered scenario, IoT devices communicate through a public channel. An adversary can eavesdrop, modify, remove, and duplicate messages transmitted in the public channel. The adversary can act as an insider to obtain secret parameters of other members to implement attacks. Though the proposed scenario and scheme are interesting, we find the scheme is flawed because of some false computations. We also find it is vulnerable to key compromise impersonation attack.

2 Review of the scheme

Let p be a large prime number, F_p be a prime finite field, E/F_p be an elliptic curve over field F_p, Z_p^* be the set of numbers $\{1, \cdots, p-1\}$, G be a base point over E/F_p. $h(\cdot)$ is a one-way hash function defined by: $\{0,1\}^* \to \{0,1\}^l$ with arbitrary length inputs and fixed-length outputs. $\triangle T$ is the maximum allowable transmission delay.

The scheme [1] consists of four phases: initial system configuration, registration and key generation, authentication and key agreement, public and private keys update. For conveniences, we now describe the related phases as follows (see Table 1).

[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 Hajian-Haghighat-Erfani key agreement scheme

User U_i	Initial System Configuration	TA
		Pick $s, K_{pri} \in Z_p^*$ and set $K_{pub} = K_{pri}G$.
	request $\longrightarrow$	Assign a unique ID to the IoT device.
		Compute $HID_x = h(ID_x \| s \| K_{pri})$.
	$\Longleftarrow$ device D_i [secure channel]	Store $\{HID_x, sG\}$ in the device.
		Publish $\{E/F_p, h(\cdot), G, K_{pub}\}$.
Device $D_i : \{HID_i, sG\}$	Registration & Key Generation	TA: $\{K_{pri}, s\}$
Pick $r_i \in Z_p^*$, compute $R_i = r_iG$,		
$MID_i = ID_i \oplus h(r_isG)$,		Compute $ID_i = MID_i \oplus h(sR_i)$,
$MAC_i = h(ID_i \| HID_i \| t_1)$,	$MID_i,\ R_i,\ MAC_i,\ t_1 \longrightarrow$ [public channel]	$HID_i = h(ID_i \| s \| K_{pri})$.
where t_1 is the timestamp.		Check if $MAC_i = h(ID_i \| HID_i \| t_1)$.
		If true, pick $e_i \in Z_p^*$,
		compute $P_i = e_iK_{pub} + R_i$,
		$f_i = (e_i + P_i)K_{pri}$,
		$Q_i = P_i + P_iK_{pub}$,
Compute		$M_i = K_{pri}Q_i, N_i = M_i + h(ID_i \| sR_i)G$,
$f_i = F_i \oplus h(ID_i \| R_i)$,	$\longleftarrow F_i,\ N_i,\ MAC_{TA}$	$F_i = f_i \oplus h(ID_i \| R_i)$,
$d_i = f_i + r_i$,		$MAC_{TA} = h(HID_i \| N_i \| F_i)$,
$M_i = N_i - h(ID_i \| r_isG)G$.		$\xi_i = e_i \oplus h(R_i \| K_{pri})$.
Check if $d_iG = Q_i$ and		Store $\{\xi_i, HID_i, R_i\}$. Publish P_i, Q_i.
$MAC_{TA} = h(HID_i \| N_i \| F_i)$.		
If true, store $\{M_i \oplus ID_i\}$.		
D_i	Authentication & Key Agreement	D_j
Pick $x \in Z_p^*$, the timestamp t_1,		
compute $TID_i = h(ID_i \| x)$,		
$\alpha_i = xQ_j, \tau_i = xG$,	$AID_i, Z_i, \tau_i, t_1 \longrightarrow$	Check if $\|t_1' - t_1\| < \triangle T$.
$AID_i = h(\alpha_i) \oplus TID_i$,		If true, compute $\alpha_i = d_j\tau_i$,
$Z_i = h(TID_i \| \tau_i \| M_iQ_j \| t_1)$.		$TID_i = h(\alpha_i) \oplus AID_i$.
		Check if $Z_i = h(TID_i \| \tau_i \| M_jQ_i \| t_1)$.
		If true, pick $y \in Z_p^*$ and
		the timestamp t_2, compute
Check if $\|t_2' - t_2\| < \triangle T$.		$\beta_j = yQ_i, \tau_j = yG$,
If true, compute $\beta_j = d_i\tau_j$,	$\longleftarrow AID_j, Z_j, \tau_j, t_2$	$TID_j = h(ID_j \| y)$,
$TID_j = h(\beta_j) \oplus AID_j$,		$AID_j = h(\beta_j) \oplus TID_j$,
$K_i = h(\alpha_i \| \beta_j \| M_iQ_j)$,		$K_j = h(\alpha_i \| \beta_j \| M_jQ_i)$,
$SK_i = h(TID_i \| TID_j \| K_i \| t_2)$.		$SK_j = h(TID_i \| TID_j \| K_j \| t_2)$,
Check if $Z_j = h(TID_j \| SK_i \| t_2)$.		$Z_j = h(TID_j \| SK_j \| t_2)$.

3 Some false computations and corrections

The scheme confuses the underlying elliptic curve group operation, which results in some false computations. In the registration and key generation phase, it specifies that

$$f_i = (e_i + P_i)K_{pri} \tag{1}$$

where $e_i \in Z_p^*$ is picked by the TA, $K_{pri} \in Z_p^*$ is the TA's private key. But

$$P_i = e_i K_{pub} + R_i = (e_i K_{pri} + r_i)G \in E/F_p \tag{2}$$

is a point over the elliptic curve. So, the notation $e_i + P_i$ makes no sense because it tries to add two different objects. The following computation

$$Q_i = P_i + P_i K_{pub} \tag{3}$$

is false, too. In fact,

$$P_i \in E/F_p,\ K_{pub} = K_{pri}G \in E/F_p \tag{4}$$

The notation $P_i K_{pub}$ is not well-defined [2]. To revise, one can specify that

$$f_i = (e_i + \hat{h}(P_i))K_{pri} \tag{1'}$$

$$Q_i = P_i + \hat{h}(P_i)K_{pub} \tag{3'}$$

where $\hat{h} : \{0,1\}^* \to Z_p^*$ is a hash function.

Correctness. We have

$$\begin{aligned} Q_i &= P_i + \hat{h}(P_i)K_{pub} = (e_i K_{pub} + R_i) + \hat{h}(P_i)K_{pub} \\ &= (e_i + \hat{h}(P_i))K_{pub} + R_i = (e_i + \hat{h}(P_i))K_{pri}G + r_i G = (f_i + r_i)G = d_i G \end{aligned}$$

In the authentication and key agreement phase, it specifies that

$$Z_i = h(TID_i \| \tau_i \| M_i Q_j \| t_1) \tag{5}$$

Note that

$$M_i = K_{pri}Q_i = K_{pri}d_i G \in E/F_p,\ Q_j = d_j G \in E/F_p \tag{6}$$

So, both are two points over the elliptic curve and the notation $M_i Q_j$ makes no sense. Likewise, the notation $M_j Q_i$ is false, too. To revise, it needs to replace them as follows

$$M_i Q_j \leftarrow d_i Q_j, \quad M_j Q_i \leftarrow d_j Q_i \tag{7}$$

In this case, $d_i Q_j = d_i d_j G = d_j d_i G = d_j Q_i$. Unfortunately, this replacement results in other security problems because the secret parameters M_i, M_j will be never invoked in the whole process.

4 Insecure against key compromise impersonation attack

In the considered model, it assumes that the adversary can compromise all network entities and obtain all temporary as well as permanent credentials (see §3.2, on page 5, [1]). As for the security against key compromise impersonation attack, it argues that (see page 18, [1]):

> *Once an adversary A gains access to private keys d_i or d_j, he is unable to get session key $SK_i = SK_j$ because the hidden parameters M_i/M_j are stored in devices as $M_i = K_{pri}Q_i$, which are extracted only when the Elliptic Curve Diffie-Hellman Problem (ECDHP) is solved.*

This is a clear self-contradiction. In fact,

$$M_i = K_{pri}Q_i = K_{pri}d_iG = d_iK_{pri}G = d_iK_{pub} \tag{8}$$

Since K_{pub} is a system public key, it is certainly available to the adversary. Therefore, the adversary can retrieve the hidden parameter M_i once d_i is compromised.

5 Conclusion

In this note, we show that the Hajian-Haghighat-Erfani key agreement scheme has some flaws. The findings could be helpful for the future works on designing such schemes.

References

[1] R. Hajian, A. Haghighat, S. Erfani: A Secure Anonymous D2D Mutual Authentication and Key Agreement Protocol for IoT, Internet of Things, 18:100493 (2022)

[2] D. Hankerson, S. Vanstone, A. Menezes : Guide to Elliptic Curve Cryptography. Springer New York, USA (2006)