Security Analysis of Two Unbalancing Pairing-free Identity-based Authenticated Key Exchange Protocols

Qingfeng Cheng[1,2], Yuting Li[1,2], Qi Jiang[3], and Xiong Li[4]
(Corresponding author: Qingfeng Cheng)

Strategic Support Force Information Engineering University[1]
Zhengzhou 450001, P. R. China
State Key Laboratory of Mathematical Engineering and Advanced Computing[2]
Zhengzhou 450001, P. R. China
School of Cyber Engineering, Xidian University[3]
Xi'an, 710071, P. R. China
School of Computer Science and Engineering, Hunan University of Science and Technology[4]
Xiangtan, 411201, P.R. China
(Email: qingfengc2008@sina.com)
(Received Feb. 13, 2019; Revised and Accepted Sept. 16, 2019; First Online Feb. 9, 2020)

Abstract

The Internet of Things plays an increasingly important role in various fields. However, there are many devices in the Internet of Things that are unbalanced in terms of computing and storage capacity, which should be given full consideration. Recently, Zhang *et al.* proposed two unbalancing pairing-free identity-based authenticated key exchange (AKE) protocols for disaster scenarios, which was claimed to achieve forward security and impersonation attack resilience. In this paper, we show that two proposed AKE protocols are lack of forward security and also cannot resist key compromise impersonation attack.

Keywords: Authenticated Key Exchange; Forward Security; Key Compromise Impersonation Attack; Pair-Free

1 Introduction

The Internet of Things has been developing rapidly in recent years, bringing a lot of convenience services to people in various fields of the society. In the environment of Internet of Things, there are a substantial number of sensors, radio frequency cards and other devices with different computing and storage capabilities. In order to ensure secure communications among these devices, we usually use authenticated key exchange (AKE) protocols [2,6–13] to generate the session keys for encrypting messages over public network.

Although there are many AKE protocols for Internet of Things, they are seldom designed for disaster scenarios. In such scenarios, secure data transmissions among unbalanced devices are very important. Recently, Zhang *et al.* [14] proposed two pairing-free identity-based AKE protocols, called UPIAP1 protocol and UPIAP2 protocol. Both of them were designed for the limited devices with unbalanced computational ability. Zhang *et al.* proved their two UPIAP protocols' security in the mBR model [2] and compared the performance with pairing-free AKE protocols in [3, 5, 12]. However, in this paper, we will analyze the security of the UPIAP1 protocol and UPIAP2 protocol, and show that both of them still exist some security flaws. In details, if the adversary can learn two parties' long-term private keys, he can recover the previous session keys. In addition, if the adversary can learn a party's secret key or partial secret key, he can impersonate the other party to cheat the party, who divulges his own long-term private key.

The remainder of this paper will firstly introduce some notations and desirable security attributes in Section 2. Then we briefly review UPIAP1 protocol and UPIAP2 protocol in Section 3. Further, Section 4 points out the weaknesses of UPIAP1 protocol and UPIAP2 protocol. Conclusion will be given in Section 5.

2 Preliminaries

This section briefly introduces some notations and security attributes in Table 1, which are used in the UPIAP1 protocol and UPIAP2 protocol. More details can refer to [14].

In general, the basic desirable attributes of secure AKE protocols include key compromise impersonation (KCI)

Table 1: Notations

Notations	Description
τ	security parameter
Z_p^*	$\{1, 2, \cdots, p-1\}$
$\mathcal{G}$	a cyclic additive group of order p, P is a generator of this group
$\mathcal{M}$	the adversary
s	Key Generation Center (KGC)'s master private key
P_{pub}	KGC's master public key, where $P_{pub} = sP$
$\hat{X}$	party who involves in the AKE protocol
$(s_{\hat{X}}, v_{\hat{X}})$	party $\hat{X}$'s long-term private key, where $s_{\hat{X}}P = R_{\hat{X}} + H_1(\hat{X} \parallel R_{\hat{X}}) \cdot P_{pub}$ and $v_{\hat{X}} \in Z_p^*$
$(R_{\hat{X}}, V_{\hat{X}})$	party $\hat{X}$'s long-term public key, where $R_{\hat{X}} = r_{\hat{X}} \cdot P$, $r_{\hat{X}} \in Z_p^*$ and $V_{\hat{X}} = v_{\hat{X}} \cdot P$
H_1	a hash function from $\{0,1\}^*$ to Z_p^*
H_2	a hash function from $\{0,1\}^*$ to $\{0,1\}^\tau$
$HMAC$	a verification hash function from $\{0,1\}^*$ to $\{0,1\}^\tau$

security, key control security and forward security [4], *etc.* In this section, we only describe some security attributes involved in the analysis of two UPIAP protocols.

- **Forward security**. If two parties' long-term private keys are compromised simultaneously, the adversary cannot recover previous session keys.
- **KCI security**. Suppose $\hat{A}$'s private key $(s_{\hat{A}}, v_{\hat{A}})$ is compromised. The adversary cannot impersonate $\hat{B}$ to cheat $\hat{A}$.
- **Partial KCI security**. Suppose $\hat{A}$'s partial key $v_{\hat{A}}$ is compromised. The adversary cannot impersonate $\hat{B}$ to cheat $\hat{A}$.

3 Review of Two UPIAP Protocols

This section describes Zhang *et al.*'s two UPIAP protocols for disaster scenarios, which are claimed to achieve forward security and impersonation attack resilience. Two UPIAP protocols include a KGC and two parties respectively, where the KGC initializes the key exchange system parameters. For the sake of brevity, we omit some unnecessary descriptions.

3.1 UPIAP1 Protocol

In this subsection, we briefly review Zhang *et al.*'s UPIAP1 protocol.

Step 1. The initiator $\hat{A}$ randomly generates a value $\hat{a} \in Z_p^*$. Then $\hat{A}$ sends the message $\hat{I}_1$ to the responder $\hat{B}$ as follows:

$$\hat{A} \rightarrow \hat{B} : \hat{I}_1 = \{R_{\hat{A}}, V_{\hat{A}}, Eph_{\hat{A}}\},$$

where $Eph_{\hat{A}} = \hat{a} + v_{\hat{A}}$.

Step 2. After receiving the message $\hat{I}_1$, $\hat{B}$ randomly generates a value $\hat{b} \in Z_p^*$ and computes $Eph_{\hat{B}} = \hat{b} + v_{\hat{B}}$. Then $\hat{B}$ computes the session key components as follows:

$$K_{\hat{B}1} = s_{\hat{B}} \cdot (T_{\hat{A}} - V_{\hat{A}}) + \hat{b}(R_{\hat{A}} + H_1(\hat{A} \parallel R_{\hat{A}}) \cdot P_{pub}),$$

$$K_{\hat{B}2} = \hat{b} \cdot (T_{\hat{A}} - V_{\hat{A}}),$$

where $T_{\hat{A}} = Eph_{\hat{A}} \cdot P$ and $T_{\hat{B}} = Eph_{\hat{B}} \cdot P$.

Finally, $\hat{B}$ sends the message $\hat{R}$ to $\hat{A}$ as follows:

$$\hat{B} \rightarrow \hat{A} : \hat{R} = \{R_{\hat{B}}, V_{\hat{B}}, T_{\hat{B}}, T_{\hat{A}}, MAC_{\hat{B}}\},$$

where $MAC_{\hat{B}} = HMAC(K_{\hat{B}1} \parallel K_{\hat{B}2}, R_{\hat{B}} \parallel V_{\hat{B}} \parallel T_{\hat{B}} \parallel T_{\hat{A}})$.

Step 3. After receiving the message $\hat{R}$, $\hat{A}$ generates the session key components as follows:

$$K_{\hat{A}1} = s_{\hat{A}} \cdot (T_{\hat{B}} - V_{\hat{B}}) + \hat{a}(R_{\hat{B}} + H_1(\hat{B} \parallel R_{\hat{B}}) \cdot P_{pub}),$$

$$K_{\hat{A}2} = \hat{a} \cdot (T_{\hat{B}} - V_{\hat{B}}).$$

Then $\hat{A}$ checks $MAC_{\hat{B}}$. If $VER(K_{\hat{A}1} \parallel K_{\hat{A}2}, R_{\hat{B}} \parallel V_{\hat{B}} \parallel Eph_{\hat{B}}, MAC_{\hat{B}})$ equals to 1, it is valid. $\hat{A}$ generates the session key $SK_{\hat{A}\hat{B}} = H_2(\hat{A} \parallel \hat{B} \parallel T_{\hat{A}} \parallel T_{\hat{B}} \parallel K_{\hat{A}1} \parallel K_{\hat{A}2})$ and sends the message $\hat{I}_2$ to $\hat{B}$:

$$\hat{A} \rightarrow \hat{B} : \hat{I}_2 = \{MAC_{\hat{A}}\},$$

where $MAC_{\hat{A}} = HMAC(K_{\hat{A}1} \parallel K_{\hat{A}2}, R_{\hat{A}} \parallel V_{\hat{A}} \parallel Eph_{\hat{A}})$.

Step 4. After receiving the message $\hat{I}_2$, $\hat{B}$ checks $MAC_{\hat{A}}$. If $VER(K_{\hat{B}1} \parallel K_{\hat{B}2}, R_{\hat{A}} \parallel V_{\hat{A}} \parallel Eph_{\hat{A}}, MAC_{\hat{A}})$ equals to 1, it is valid. $\hat{B}$ generates the session key $SK_{\hat{B}\hat{A}} = H_2(\hat{A} \parallel \hat{B} \parallel T_{\hat{A}} \parallel T_{\hat{B}} \parallel K_{\hat{B}1} \parallel K_{\hat{B}2})$.

If $VER(K_{\hat{B}1} \parallel K_{\hat{B}2}, R_{\hat{A}} \parallel V_{\hat{A}} \parallel Eph_{\hat{A}}, MAC_{\hat{A}})$ equals to 0, it is invalid. $\hat{B}$ aborts the session.

3.2 UPIAP2 Protocol

In this subsection, we briefly review Zhang *et al.*'s UPIAP2 protocol.

Step 1. The initiator $\hat{A}$ randomly generates a value $\hat{a} \in Z_p^*$. Then $\hat{A}$ sends the message $\hat{I}_1$ to the responder $\hat{B}$ as follows:

$$\hat{A} \rightarrow \hat{B} : \hat{I}_1 = \{R_{\hat{A}}, V_{\hat{A}}, T_{\hat{A}}\},$$

where $Eph_{\hat{A}} = \hat{a} + v_{\hat{A}}$, $T_{\hat{A}} = Eph_{\hat{A}} \cdot P$.

Step 2. After receiving the message $\hat{I}_1$, $\hat{B}$ randomly generates a value $\hat{b} \in Z_p^*$ and computes $Eph_{\hat{B}} = \hat{b} + v_{\hat{B}}$. Then $\hat{B}$ computes the session key components as follows:

$$K_{\hat{B}1} = s_{\hat{B}} \cdot (T_{\hat{A}} - V_{\hat{A}}) + \hat{b}(R_{\hat{A}} + H_1(\hat{A} \parallel R_{\hat{A}}) \cdot P_{pub}),$$

$$K_{\hat{B}2} = \hat{b} \cdot (T_{\hat{A}} - V_{\hat{A}}).$$

Finally, $\hat{B}$ sends the message $\hat{R}$ to $\hat{A}$ as follows:

$$\hat{B} \rightarrow \hat{A} : \hat{R} = \{R_{\hat{B}}, V_{\hat{B}}, Eph_{\hat{B}}, MAC_{\hat{B}}\},$$

where $MAC_{\hat{B}} = HMAC(K_{\hat{B}1} \parallel K_{\hat{B}2}, R_{\hat{B}} \parallel V_{\hat{B}} \parallel Eph_{\hat{B}})$.

Step 3. After receiving the message $\hat{R}$, $\hat{A}$ generates the session key components as follows:

$$K_{\hat{A}1} = s_{\hat{A}} \cdot (T_{\hat{B}} - V_{\hat{B}}) + \hat{a}(R_{\hat{B}} + H_1(\hat{B} \parallel R_{\hat{B}}) \cdot P_{pub}),$$

$$K_{\hat{A}2} = \hat{a} \cdot (T_{\hat{B}} - V_{\hat{B}}).$$

Then $\hat{A}$ checks $MAC_{\hat{B}}$. If $VER(K_{\hat{A}1} \parallel K_{\hat{A}2}, R_{\hat{B}} \parallel V_{\hat{B}} \parallel Eph_{\hat{B}}, MAC_{\hat{B}})$ equals to 1, it is valid. $\hat{A}$ generates the session key $SK_{\hat{A}\hat{B}} = H_2(\hat{A} \parallel \hat{B} \parallel T_{\hat{A}} \parallel T_{\hat{B}} \parallel K_{\hat{A}1} \parallel K_{\hat{A}2})$ and sends the message $\hat{I}_2$ to $\hat{B}$:

$$\hat{A} \rightarrow \hat{B} : \hat{I}_2 = \{T_{\hat{B}}, MAC_{\hat{A}}\},$$

where $MAC_{\hat{A}} = HMAC(K_{\hat{A}1} \parallel K_{\hat{A}2}, R_{\hat{A}} \parallel V_{\hat{A}} \parallel Eph_{\hat{A}})$.

If $VER(K_{\hat{A}1} \parallel K_{\hat{A}2}, R_{\hat{B}} \parallel V_{\hat{B}} \parallel Eph_{\hat{B}}, MAC_{\hat{B}})$ equals to 0, it is invalid. $\hat{A}$ aborts the session.

Step 4. After receiving the message $\hat{I}_2$, $\hat{B}$ checks $MAC_{\hat{A}}$. If $VER(K_{\hat{B}1} \parallel K_{\hat{B}2}, R_{\hat{A}} \parallel V_{\hat{A}} \parallel Eph_{\hat{A}}, MAC_{\hat{A}})$ equals to 1, it is valid. $\hat{B}$ generates the session key $SK_{\hat{B}\hat{A}} = H_2(\hat{A} \parallel \hat{B} \parallel T_{\hat{A}} \parallel T_{\hat{B}} \parallel K_{\hat{B}1} \parallel K_{\hat{B}2})$.

If $VER(K_{\hat{B}1} \parallel K_{\hat{B}2}, R_{\hat{A}} \parallel V_{\hat{A}} \parallel Eph_{\hat{A}}, MAC_{\hat{A}})$ equals to 0, it is invalid. $\hat{B}$ aborts the session.

4 Analysis of Two UPIAP Protocols

This section will analyze Zhang *et al.*'s two UPIAP protocols and point out security flaws of two UPIAP protocols. Since UPIAP1 protocol and UPIAP2 protocol are similar in structure, we only describe the analysis of UPIAP1 protocol.

4.1 Analysis of Forward Security

In the UPIAP1 protocol, Zhang *et al.* claimed that the adversary could not obtain previous session keys, even if the adversary could get $\hat{A}$ and $\hat{B}$'s long-term private keys by stolen device attack. However, we carefully analyze the UPIAP1 protocol, and prove this protocol without forward security.

The adversary $\mathcal{M}$ can obtain $\hat{A}$'s secret key $(s_{\hat{A}}, v_{\hat{A}})$. Since the public ephemeral message $Eph_{\hat{A}} = \hat{a} + v_{\hat{A}}$, $\mathcal{M}$ can compute the value of $\hat{a}$ through the public ephemeral message $Eph_{\hat{A}}$. Further, $\mathcal{M}$ can use $s_{\hat{A}}$ and $v_{\hat{A}}$ to compute the session key components $K_{\hat{A}1}$ and $K_{\hat{A}2}$. Finally, $\mathcal{M}$ can use $K_{\hat{A}1}$ and $K_{\hat{A}2}$ to recover the previous session key $SK_{\hat{A}\hat{B}} = H_2(\hat{A} \parallel \hat{B} \parallel T_{\hat{A}} \parallel T_{\hat{B}} \parallel K_{\hat{A}1} \parallel K_{\hat{A}2})$, because $\hat{A}, \hat{B}, T_{\hat{A}}$ and $T_{\hat{B}}$ are also public messages.

Similarly, the adversary $\mathcal{M}$ can mount the attack to the UPIAP2 protocol successfully. So two proposed protocols are lack of forward security.

4.2 KCI Attack

The key compromise impersonation (KCI) attack resilience is a basic attribute for AKE protocols. In this subsection, we will prove that the UPIAP1 protocol cannot resist KCI attack. We assume the adversary $\mathcal{M}$ has obtained party $\hat{A}$'s secret key $(s_{\hat{A}}, v_{\hat{A}})$. Then the adversary $\mathcal{M}$ impersonates party $\hat{B}$ to cheat $\hat{A}$. The KCI attack's details are as follows.

Step 1. The initiator $\hat{A}$ randomly generates a value $\hat{a} \in Z_p^*$. Then $\hat{A}$ sends the message $\hat{I}_1$ to the responder $\hat{B}$ as follows:

$$\hat{A} \rightarrow \hat{B} : \hat{I}_1 = \{R_{\hat{A}}, V_{\hat{A}}, Eph_{\hat{A}}\},$$

where $Eph_{\hat{A}} = \hat{a} + v_{\hat{A}}$.

Step 2. The adversary $\mathcal{M}$ intercepts the message $\hat{I}_1$, $\mathcal{M}$ randomly generates a value $\hat{m} \in Z_p^*$ and computes $T_{\mathcal{M}} = \hat{m} \cdot P + V_{\hat{B}}$. Since $\mathcal{M}$ has $s_{\hat{A}}$ and $v_{\hat{A}}$, $\mathcal{M}$ can compute $\hat{a}$ from $Eph_{\hat{A}}$ and $v_{\hat{A}}$. Then $\mathcal{M}$ computes the session key components as follows:

$$K_{\mathcal{M}1} = s_{\hat{A}} \cdot (T_{\mathcal{M}} - V_{\hat{B}}) + \hat{a}(R_{\hat{B}} + H_1(\hat{B} \parallel R_{\hat{B}}) \cdot P_{pub}),$$

$$K_{\mathcal{M}2} = \hat{a} \cdot (T_{\mathcal{M}} - V_{\hat{B}}),$$

Finally, the adversary $\mathcal{M}$ impersonates $\hat{B}$ to send $\hat{R}_{\mathcal{M}}$ to $\hat{A}$ as follows:

$$\hat{B}(\mathcal{M}) \rightarrow \hat{A} : \hat{R}_{\mathcal{M}} = \{R_{\hat{B}}, V_{\hat{B}}, T_{\mathcal{M}}, T_{\hat{A}}, MAC_{\mathcal{M}}\},$$

where $MAC_{\mathcal{M}} = HMAC(K_{\mathcal{M}1} \parallel K_{\mathcal{M}2}, R_{\hat{B}} \parallel V_{\hat{B}} \parallel T_{\mathcal{M}} \parallel T_{\hat{A}})$.

Step 3. After receiving the message $\hat{R}_{\mathcal{M}}$, $\hat{A}$ computes the session key components as follows:

$$K_{\hat{A}1} = s_{\hat{A}} \cdot (T_{\mathcal{M}} - V_{\hat{B}}) + \hat{a}(R_{\hat{B}} + H_1(\hat{B} \parallel R_{\hat{B}}) \cdot P_{pub}),$$

$$K_{\hat{A}2} = \hat{a} \cdot (T_{\mathcal{M}} - V_{\hat{B}}).$$

Then $\hat{A}$ checks $MAC_{\mathcal{M}}$. If $VER(K_{\hat{A}1} \parallel K_{\hat{A}2}, R_{\hat{B}} \parallel V_{\hat{B}} \parallel Eph_{\hat{B}}, MAC_{\mathcal{M}})$ equals to 1, it is valid. $\hat{A}$ generates the session key $SK_{\hat{A}\hat{B}} = H_2(\hat{A} \parallel \hat{B} \parallel T_{\hat{A}} \parallel T_{\mathcal{M}} \parallel K_{\hat{A}1} \parallel K_{\hat{A}2})$ and sends the message $\hat{I}_2$ to $\hat{B}$:

$$\hat{A} \rightarrow \hat{B} : \hat{I}_2 = \{MAC_{\hat{A}}\},$$

where $MAC_{\hat{A}} = HMAC(K_{\hat{A}1} \parallel K_{\hat{A}2}, R_{\hat{A}} \parallel V_{\hat{A}} \parallel Eph_{\hat{A}})$.

Step 4. After intercepting the message $\hat{I}_2$, the adversary $\mathcal{M}$ also computes the session key $SK_{\mathcal{M}\hat{A}} = H_2(\hat{A} \parallel \hat{B} \parallel T_{\hat{A}} \parallel T_{\mathcal{M}} \parallel K_{\mathcal{M}1} \parallel K_{\mathcal{M}2})$.

Since we have $K_{\hat{A}1} = K_{\mathcal{M}1}$ and $K_{\hat{A}2} = K_{\mathcal{M}2}$, it means that the adversary $\mathcal{M}$ can pass $\hat{A}$'s verification successfully and generate the same session key as $\hat{A}$.

Similarly, the adversary $\mathcal{M}$ can mount KCI attack to the UPIAP2 protocol successfully.

4.3 Partial KCI Attack

In this subsection, we will prove that the UPIAP1 protocol cannot resist partial KCI attack either. We assume the adversary $\mathcal{M}$ has only obtained party $\hat{A}$'s partial secret key $v_{\hat{A}}$. Then the adversary $\mathcal{M}$ impersonates party $\hat{B}$ to cheat $\hat{A}$. The partial KCI attack's details are as follows.

Step 1. The initiator $\hat{A}$ randomly generates a value $\hat{a} \in Z_p^*$. Then $\hat{A}$ sends the message $\hat{I}_1$ to the responder $\hat{B}$ as follows:

$$\hat{A} \rightarrow \hat{B} : \hat{I}_1 = \{R_{\hat{A}}, V_{\hat{A}}, Eph_{\hat{A}}\},$$

where $Eph_{\hat{A}} = \hat{a} + v_{\hat{A}}$.

Step 2. The adversary $\mathcal{M}$ intercepts the message $\hat{I}_1$, $\mathcal{M}$ randomly generates a value $\hat{m} \in Z_p^*$ and computes $T_{\mathcal{M}} = \hat{m} \cdot P + V_{\hat{B}}$. Since $\mathcal{M}$ has obtained $v_{\hat{A}}$, $\mathcal{M}$ can compute $\hat{a}$ from $Eph_{\hat{A}}$ and $v_{\hat{A}}$. Then $\mathcal{M}$ computes the session key components as follows:

$$\begin{aligned} K_{\mathcal{M}1} &= \hat{m} \cdot (R_{\hat{A}} + H_1(\hat{A} \parallel R_{\hat{A}}) \cdot P_{pub}) + \\ &\quad \hat{a}(R_{\hat{B}} + H_1(\hat{B} \parallel R_{\hat{B}}) \cdot P_{pub}), \\ K_{\mathcal{M}2} &= \hat{a} \cdot (T_{\mathcal{M}} - V_{\hat{B}}). \end{aligned}$$

Finally, the adversary $\mathcal{M}$ impersonates $\hat{B}$ and sends $\hat{R}_{\mathcal{M}}$ to $\hat{A}$ as follows:

$$\hat{B}(\mathcal{M}) \rightarrow \hat{A} : \hat{R}_{\mathcal{M}} = \{R_{\hat{B}}, V_{\hat{B}}, T_{\mathcal{M}}, T_{\hat{A}}, MAC_{\mathcal{M}}\},$$

where $MAC_{\mathcal{M}} = HMAC(K_{\mathcal{M}1} \parallel K_{\mathcal{M}2}, R_{\hat{B}} \parallel V_{\hat{B}} \parallel T_{\mathcal{M}} \parallel T_{\hat{A}})$.

Step 3. After receiving the message $\hat{R}_{\mathcal{M}}$, $\hat{A}$ computes the session key components as follows:

$$K_{\hat{A}1} = s_{\hat{A}} \cdot (T_{\mathcal{M}} - V_{\hat{B}}) + \hat{a}(R_{\hat{B}} + H_1(\hat{B} \parallel R_{\hat{B}}) \cdot P_{pub}),$$

$$K_{\hat{A}2} = \hat{a} \cdot (T_{\mathcal{M}} - V_{\hat{B}}).$$

Then $\hat{A}$ checks $MAC_{\mathcal{M}}$. We have

$$\begin{aligned} K_{\mathcal{M}1} &= \hat{m} \cdot (R_{\hat{A}} + H_1(\hat{A} \parallel R_{\hat{A}}) \cdot P_{pub}) \\ &\quad + \hat{a}(R_{\hat{B}} + H_1(\hat{B} \parallel R_{\hat{B}}) \cdot P_{pub}) \\ &= \hat{m}s_{\hat{A}}P + \hat{a}s_{\hat{B}}P \\ &= s_{\hat{A}} \cdot (T_{\mathcal{M}} - V_{\hat{B}}) \\ &\quad + \hat{a}(R_{\hat{B}} + H_1(\hat{B} \parallel R_{\hat{B}}) \cdot P_{pub}) \\ &= K_{\hat{A}1}, \\ K_{\mathcal{M}2} &= \hat{a} \cdot (T_{\mathcal{M}} - V_{\hat{B}}) = \hat{a}\hat{m}P \\ &= \hat{m}\hat{a}P \\ &= \hat{a} \cdot (T_{\mathcal{M}} - V_{\hat{B}}) \\ &= K_{\hat{A}2}. \end{aligned}$$

So $VER(K_{\hat{A}1} \parallel K_{\hat{A}2}, R_{\hat{B}} \parallel V_{\hat{B}} \parallel Eph_{\hat{B}}, MAC_{\mathcal{M}})$ equals to 1, it is valid. $\hat{A}$ generates the session key $SK_{\hat{A}\hat{B}} = H_2(\hat{A} \parallel \hat{B} \parallel T_{\hat{A}} \parallel T_{\mathcal{M}} \parallel K_{\hat{A}1} \parallel K_{\hat{A}2})$ and sends the message $\hat{I}_2$ to $\hat{B}$:

$$\hat{A} \rightarrow \hat{B} : \hat{I}_2 = \{MAC_{\hat{A}}\},$$

where $MAC_{\hat{A}} = HMAC(K_{\hat{A}1} \parallel K_{\hat{A}2}, R_{\hat{A}} \parallel V_{\hat{A}} \parallel Eph_{\hat{A}})$.

Step 4. After intercepting the message $\hat{I}_2$, the adversary $\mathcal{M}$ also computes the session key $SK_{\mathcal{M}\hat{A}} = H_2(\hat{A} \parallel \hat{B} \parallel T_{\hat{A}} \parallel T_{\mathcal{M}} \parallel K_{\mathcal{M}1} \parallel K_{\mathcal{M}2})$.

Since we have $K_{\hat{A}1} = K_{\mathcal{M}1}$ and $K_{\hat{A}2} = K_{\mathcal{M}2}$, it means that the adversary $\mathcal{M}$ can generate the same session key as $\hat{A}$.

Similarly, the adversary $\mathcal{M}$ can mount partial KCI attack to the UPIAP2 protocol successfully.

5 Conclusion

Secure communication is a vital point in disaster environment, and encryption is the basic guarantees for communication messages. There have existed many AKE protocols to generate session keys for encryption. Especially, pairing-free identity-based AKE protocols are more adapt for Internet of Things to generate these session keys. In this paper, we analyzes the UPIAP1 protocol and UPIAP2 protocol, which are two pairing-free identity-based AKE protocols proposed by Zhang *et al.* in 2019. The analysis results show that two UPIAP protocols cannot obtain the attribute of forward security, or resist KCI attack as well as partial KCI attack. The main reason for this situation is that there are some security flaws in the misusage of ephemeral key and long-term private key. For designing better protocols to remedy these flaws, we recommend to use the method in [1, 12].

Acknowledgments

The authors would like to thank Prof. Min-Shiang Hwang and the anonymous referees for their helpful comments.

This work was supported by the National Natural Science Foundation of China (No. 61872449).

References

[1] S. Bala, G. Sharma, A. Verma, "PF-ID-2PAKA: pairing free identity-based two-party authenticated key agreement protocol for wireless sensor networks," *Wireless Personal Communications*, vol. 87, no. 3, pp. 995-1012, 2016.

[2] M. Bellare, D. Pointcheval, P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Proceedings of Advances in Cryptology*, pp. 139-155, 2000.

[3] X. Cao, W. Kou, X. Du, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges," *Information Sciences*, vol. 180, no. 15, pp. 2895-2903, 2010.

[4] Q. Cheng, X. Zhang, "Comments on privacy-preserving Yoking proof with key exchange in the three-party setting," *International Journal of Network Security*, vol. 21, no. 2, pp. 355-358, 2019.

[5] L. Dang, J. Xu, X. Cao, H. Li, J. Chen, Y. Zhang, X. Fu, "Efficient identity-based authenticated key agreement protocol with provable security for vehicular ad hoc networks," *International Journal of Distributed Sensor Networks*, vol. 14, no. 4, pp. 1-16, 2018.

[6] C. Kudla, K. G. Paterson, "Modular security proofs for key agreement protocols," in *Proceedings of Advances in Cryptology*, pp. 549-565, 2005.

[7] B. LaMacchia, K. Lauter, A. Mityagin, "Stronger security of authenticated key exchange," in *Proceedings of ProvSec, First International Cconference on Provable Security*, pp. 1-16, 2007.

[8] C. C. Lee, M. S. Hwang, L. H. Li, "A new key authentication scheme based on discrete logarithms", *Applied Mathematics and Computation*, vol. 139, no. 2, pp. 343-349, July 2003.

[9] I. C. Lin, C. C. Chang, M. S. Hwang, "Security enhancement for the simple authentication key agreement algorithm", in *Proceedings 24th Annual International Computer Software and Applications Conference (COMPSAC'00)*, 2000.

[10] C. Ling, S. Chen, M. Hwang, "Cryptanalysis of Tseng-Wu group key exchange protocol," *International Journal of Network Security*, vol. 18, no. 3, pp. 590-593, 2016.

[11] S. Nathani, B. Tripathi, S. Khatoon, "A dynamic ID based authenticated group key agreement protocol from pairing," *International Journal of Network Security*, vol. 21, no. 4, pp. 582-591, 2019.

[12] L. Ni, G. Chen, J. Li, Y. Hao, "Strongly secure identity-based authenticated key agreement protocols without bilinear pairings," *Information Sciences*, vol. 367, no. 11, pp. 176-193, 2016.

[13] G. R. Thomas, P. Armstrong, A. Boulgakov, A. Roscoe, "FDR3: A modern refinement checker for CSP," in *Proceedings of Tools and Algorithms for the Construction and Analysis of Systems*, pp. 187-201, 2014.

[14] J. Zhang, X. Huang, W. Wang, Y. Yue, "Unbalancing pairing-free identity-based authenticated key exchange protocols for disaster scenarios," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 878-890, 2019.

Biography

Qingfeng Cheng received his B. A. degree in 2000 and M. S. degree in 2004 from National University of Defense Technology, and Ph.D. degree in 2011 from Zhengzhou Information Science and Technology Institute. He is now an associate professor in the State Key Laboratory of Mathematical Engineering and Advanced Computing. His research interests include cryptography and information security.

Yuting Li is a graduate student in the State Key Laboratory of Mathematical Engineering and Advanced Computing. Her main research interests include cryptography, edge computing and information security.

Qi Jiang received the B. S. degree in computer science from Shaanxi Normal University in 2005 and Ph.D. degree in computer science from Xidian University in 2011. He is now a professor at School of Cyber Engineering, Xidian University. His research interests include security protocols, wireless network security, cloud security, *etc.*

Xiong Li received the Ph.D. degree in computer science and technology from the Beijing University of Posts and Telecommunications, Beijing, China, in 2012. He is currently an associate professor with the School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China. He has authored over 100 referred papers. His current research interests include cryptography and information security. He was a recipient of the 2015 Journal of Network and Computer Applications Best Research Paper Award.