LETTER International Journal of Network Security, Vol.21, No.2, PP.355-358, Mar. 2019 (DOI: 10.6633/IJNS.201903_21(2).20) 355

Comments on Privacy-Preserving Yoking Proof with Key Exchange in the Three-Party Setting

Qingfeng Cheng and Xinglong Zhang

(Corresponding author: Qingfeng Cheng)

State Key Laboratory of Mathematical Engineering and Advanced Computing

Zhengzhou, Henan Province 450001, China

(Email: qingfengc2008@sina.com)

(Received Oct. 1, 2017; revised and accepted Mar. 27, 2018)

Abstract

In 2017, Tian, Yang and Mu presented a new three-party key exchange protocol YPKE in radio frequency identification environment, which is based on the HMQV protocol. They claimed that the proposed YPKE protocol in the three-party setting meets user privacy and session key security. In this comment, we point out that the YPKE protocol still has some weaknesses. Our results show that the proposed YPKE protocol cannot provide perfect forward secrecy, and also cannot resist impersonation attack. At the same time, the YPKE protocol is lack of the security of ephemeral private key leakage and unknown key-share, which the original HMQV protocol can achieve.

Keywords: Cryptanalysis; Ephemeral Private Key Leakage Attack; HMQV Protocol; Key Exchange; Key Compromise Impersonation Attack; Perfect Forward Secrecy

1 Introduction

With the rise in technology, radio frequency identification (RFID) protocols [1–3, 10, 13, 14] have become essential components in the Internet of Things (IoT) environment. Usually, in a RFID protocol, the session key to encrypt communication messages among the reader(or server) and the tags (or users) is needed. Key exchange (KE), which can generate the session key, is a fundamental building block in open network. There are many famous KE protocols in the literature, such as MQV protocol [4, 5, 9], HMQV protocol [7] and NAXOS protocol [8].

Recently, Tian, Yang and Mu [12] presented a novel key exchange protocol, called YPKE protocol. The YPKE protocol using yoking proof [6] could generate a common session key among the reader(or server) and the tags (or users). The design of the YPKE protocol was based on the HMQV protocol and Schnorr signature [11]. However, in contrast to the original HMQV protocol, the YPKE protocol needs three round and involves three parties, i.e. a server and two users. In this comment, we will point out that the YPKE protocol exists some weaknesses. We show that the YPKE protocol is lack of perfect forward secrecy, and cannot resist insider impersonation attack, unknown key-share attack and ephemeral private key leakage attack, which the original HMQV protocol can resist.

The remainder of this comment will firstly introduce the original YPKE protocol in Section 2. Then, Section 3 points out the weaknesses of the YPKE protocol. Conclusion will be given in Section 4.

2 Review of the YPKE Protocol

Here, we briefly review the YPKE protocol proposed by Tian *et al.* in 2017. For more details, refer to [12].

Table 1: The notations

Notations	Description
$\mathcal{S}$	the reader/server
TX	a tag/user
τ	security parameter
$\mathcal{G}$	a cyclic additive group of order q, where $\|q\| = \tau$ is a big prime, g is a generator of this group
$SPK_{\mathcal{S}}/SSK_{\mathcal{S}}$	the server's public/secret key, where $SPK_{\mathcal{S}} = (SPK_{\mathcal{S}1}, SPK_{\mathcal{S}2})$ $SSK_{\mathcal{S}} = (SSK_{\mathcal{S}1}, SSK_{\mathcal{S}2})$
EPK_{TX}/ESK_{TX}	TX's ephemeral public/secret key
PK_{TX}/SK_{TX}	TX's public/secret key, where $PK_{TX} = (PK_{TX1}, PK_{TX2})$ $SK_{TX} = (SK_{TX1}, SK_{TX2})$
H_1	a hash function used in the HMQV from $\{0,1\}^*$ to $\{0,1\}^l$
H_1'	a hash function from $\{0,1\}^*$ to $\{0,1\}^\tau$
H_2	a hash function from $\{0,1\}^*$ to Z_q
H_3	a hash function from $\mathcal{G}$ to $\{0,1\}^\tau$
ENC	encryption function

2.1 The Description of YPKE Protocol

In this subsection, we describe the YPKE protocol shown in Figure 1, which needs five step.

1) Server $\mathcal{S}$ sends the key $SPK_{\mathcal{S}}$ to user TA and TB. This step is the same with the original YPKE protocol.

2) Upon receiving the key $SPK_{\mathcal{S}}$, TA and TB respectively send the message (EPK_{TA}, C_{TA}) and (EPK_{TB}, C_{TB}), where $C_{TA} = ENC_{SPK_{\mathcal{S}2}}(PK_{TA})$, $C_{TB} = ENC_{SPK_{\mathcal{S}2}}(PK_{TB})$, $EPK_{TA} = g^{ESK_{TA}}$ and $EPK_{TB} = g^{ESK_{TB}}$.

3) Upon receiving (EPK_{TA}, C_{TA}) and (EPK_{TB}, C_{TB}), $\mathcal{S}$ uses the key $SSK_{\mathcal{S}2}$ to obtain PK_{TA} and PK_{TB}. Then $\mathcal{S}$ sends (c, EPK_{TB}, C'_{TA}) to user TA and (c, EPK_{TA}, C'_{TB}) to user TB, where $C'_{TA} = ENC_{PK_{TA2}}(PK_{TB1})$ and $C'_{TB} = ENC_{PK_{TB2}}(PK_{TA1})$.

4) User TA decrypts $ENC_{PK_{TA2}}(PK_{TB1})$ to obtain PK_{TB1} and uses the HMQV method to compute K_{TATB}. Then TA computes $Y = g^y$, where $y = H_2(K_{TATB}||c)$, and computes $T_{TA} = g^{t_{TA}}$, where $t_{TA} \in_R Z_q$. Further, user TA computes the signature $Sig_{TA} = t_{TA} + e_{TA}SK_{TA1}$, where $e_{TA} = H_2(T_{TA}||EPK_{TA}||PK_{TB1}||C_{TA}||SPK_{\mathcal{S}1}^{ESK_{TA}}||c||Y)$. Finally, TA computes the session key $FSK = H_3(SPK_{\mathcal{S}1}^{y})$ and sends (Sig_{TA}, T_{TA}, Y) to server $\mathcal{S}$. Similarly, user TB also computes the session key $FSK = H_3(SPK_{\mathcal{S}1}^{y})$ and sends (Sig_{TB}, T_{TB}, Y) to server $\mathcal{S}$.

5) Server $\mathcal{S}$ verifies $g^{Sig_{TA}} = T_{TA} \cdot PK_{TA1}^{e_{TA}}$ and $g^{Sig_{TB}} = T_{TB} \cdot PK_{TB1}^{e_{TB}}$. If two equations are right at the same time. Then $\mathcal{S}$ computes the session key $FSK = H_3(Y^{SSK_{\mathcal{S}1}})$. Otherwise, $\mathcal{S}$ aborts the session.

3 Analysis of the YPKE Protocol

In this section, we firstly review some of the security attributes of the KE protocols, and then provide our analysis.

Perfect Forward Secrecy: A user's private key leakage does not compromise the security of session keys generated by this user before the leakage happened.

Insider Impersonation Attack: A user or the server, which involves in the protocol, is malicious, and impersonates another user (or server) to cheat the legal server (or user).

Unknown Key-Share Attack: The adversary M, can corrupt any user, mount the attack between two honest users A and B. At the end of a session, user A convinces that he has shared the session key with user B. However, user B thinks that she has shared the session key with corrupted user C.

Ephemeral Private Key Leakage Attack: The adversary learns the ephemeral private key, and uses it to compute the session key.

3.1 The Lack of Perfect Forward Secrecy

Tian *et al.* claimed that the adversary could not make corrupt queries to the server in their model. However, we think that it is not a reasonable assumption. In the YPKE protocol, there are three parties, a server and two users, whose private key and public key are independent. If the adversary can make queries to two users, he should also make queries to the server.

Since the common session key is $FSK = H_3(Y^{SSK_{\mathcal{S}1}})$, the adversary learning the server's private key $SSK_{\mathcal{S}} = (SSK_{\mathcal{S}1}, SSK_{\mathcal{S}2})$ can use the public message Y to achieve $FSK = H_3(Y^{SSK_{\mathcal{S}1}})$ easily. It means that the YPKE protocol cannot achieve the property of perfect forward secrecy.

3.2 The Description of Insider Impersonation Attack

In the original YPKE protocol, the server does not verify the identity of two users in the first round communications, so a malicious user can cheat the server successfully. Here, we assume that the user TB is a malicious user. He first fabricates a user TA^* with public key PK_{TA^*} and private key SK_{TA^*}.

1) Server $\mathcal{S}$ sends the key $SPK_{\mathcal{S}}$ to the user TA and the user TB. However, the user TB intercepts the message for the user TA. It means that the user TA even does not know the existence of the session.

2) Upon receiving the key $SPK_{\mathcal{S}}$, TA^*, who is impersonated by TB, and TB respectively send the message (EPK_{TA^*}, C_{TA^*}) and (EPK_{TB}, C_{TB}), where $C_{TA^*} = ENC_{SPK_{\mathcal{S}2}}(PK_{TA^*})$, $C_{TB} = ENC_{SPK_{\mathcal{S}2}}(PK_{TB})$, $EPK_{TA^*} = g^{ESK_{TA^*}}$ and $EPK_{TB} = g^{ESK_{TB}}$.

3) Upon receiving (EPK_{TA^*}, C_{TA^*}) and (EPK_{TB}, C_{TB}), the server $\mathcal{S}$ uses the private key $SSK_{\mathcal{S}2}$ to obtain public key PK_{TA^*} and PK_{TB} respectively. Then the server $\mathcal{S}$ sends the message (c, EPK_{TB}, C'_{TA^*}) to the user TA^* and (c, EPK_{TA^*}, C'_{TB}) to the user TB, where $C'_{TA^*} = ENC_{PK_{TA2^*}}(PK_{TB1})$ and $C'_{TB} = ENC_{PK_{TB2}}(PK_{TA1^*})$.

4) Upon intercepting the message (c, EPK_{TB}, C'_{TA^*}) and receiving the message (c, EPK_{TA^*}, C'_{TB}), user TB randomly chooses a value $y \in_R Z_q$. Then user TB computes $Y = g^y$ and $T_{TA^*} = g^{t_{TA^*}}$, where $t_{TA^*} \in_R Z_q$. Further, user TB computes the signature $Sig_{TA^*} = t_{TA^*} + e_{TA^*}SK_{TA1^*}$, where $e_{TA^*} = H_2(T_{TA^*}||EPK_{TA^*}||PK_{TB1}||C_{TA^*}||SPK_{\mathcal{S}1}^{ESK_{TA^*}}||c ||Y)$. Finally, user TB computes the final session

TA	$\mathcal{S}$	TB
$\xleftarrow{SPK_{\mathcal{S}}}$		$\xrightarrow{SPK_{\mathcal{S}}}$
$\xrightarrow{(EPK_{TA}, C_{TA})}$		$\xleftarrow{(EPK_{TB}, C_{TB})}$
$\xleftarrow{(c, EPK_{TB}, C'_{TA})}$		$\xrightarrow{(c, EPK_{TA}, C'_{TB})}$
$\xrightarrow{(Sig_{TA}, T_{TA}, Y)}$		$\xleftarrow{(Sig_{TB}, T_{TB}, Y)}$

$$TB : FSK = H_3(SPK_{\mathcal{S}1}^y)$$
$$TA : FSK = H_3(SPK_{\mathcal{S}1}^y)$$
$$\mathcal{S} : FSK = H_3(Y^{SSK_{\mathcal{S}1}})$$

Figure 1: The YPKE protocol

key $FSK = H_3(SPK_{\mathcal{S}1}^y)$ and impersonates the user TA to send $(Sig_{TA^*}, T_{TA^*}, Y)$ to server $\mathcal{S}$. Similarly, user TB also sends (Sig_{TB}, T_{TB}, Y) to the server $\mathcal{S}$.

5) Server $\mathcal{S}$ verifies $g^{Sig_{TA^*}} = T_{TA^*} \cdot PK_{TA1^*}^{e_{TA^*}}$ and $g^{Sig_{TB}} = T_{TB} \cdot PK_{TB1}^{e_{TB}}$. If two equations are right at the same time. Then $\mathcal{S}$ computes the session key $FSK = H_3(Y^{SSK_{S1}})$. Otherwise, $\mathcal{S}$ aborts the session.

Now, the session is finished. The server will think that he has shared the common session key with user TA and user TB. However, the user TA does not know the existence of the session completely. So the malicious user TB has successfully cheated the server in the session.

3.3 The Description of Unknown Key-Share Attack

In the original YPKE protocol, the user does not verify the identity of the server, so a malicious user can cheat the other user successfully. Here, we assume that the user TB is a malicious user. He can cheat the user TA, who thinks that she has shared a common session key with the server and the user TB. However, in fact, the server even does not know the existence of the session.

1) The user TB learns the server $\mathcal{S}$'s public key $SPK_{\mathcal{S}}$ and user TA's public key PK_{TA} from other sessions. Then he can impersonate the server to send $\mathcal{S}$'s public key $SPK_{\mathcal{S}}$ to user TA.

2) Upon receiving the key $SPK_{\mathcal{S}}$, TA sends the message (EPK_{TA}, C_{TA}) to the server $\mathcal{S}$, where $C_{TA} = ENC_{SPK_{\mathcal{S}2}}(PK_{TA})$ and $EPK_{TA} = g^{ESK_{TA}}$.

3) The user TB intercepts the message (EPK_{TA}, C_{TA}). Then he impersonates the server $\mathcal{S}$ and sends (c, EPK_{TB}, C'_{TA}) to the user TA, where $C'_{TA} = ENC_{PK_{TA2}}(PK_{TB1})$.

4) User TA decrypts $ENC_{PK_{TA2}}(PK_{TB1})$ to obtain PK_{TB1} and uses the HMQV method to compute K_{TATB}. Then user TA computes $Y = g^y$, where $y = H_2(K_{TATB}||c)$, and computes $T_{TA} = g^{t_{TA}}$, where $t_{TA} \in_R Z_q$. Further, user TA computes the signature $Sig_{TA} = t_{TA} + e_{TA}SK_{TA1}$, where $e_{TA} = H_2(T_{TA}||EPK_{TA}||PK_{TB1}||C_{TA}||SPK_{\mathcal{S}1}^{ESK_{TA}}||c||Y)$. Finally, TA computes the session key $FSK = H_3(SPK_{\mathcal{S}1}^y)$ and sends (Sig_{TA}, T_{TA}, Y) to the server $\mathcal{S}$.

5) Upon intercepting the message (Sig_{TA}, T_{TA}, Y), the user TB can compute the session key $FSK = H_3(SPK_{\mathcal{S}1}^y)$ and finish the session.

When the session is finished, the user TA will think that he has shared the session key with the server $\mathcal{S}$ and the user TB. In contrast, the server $\mathcal{S}$ does not know the existence of the session. So the malicious user TB has successfully cheated the user TA in the session. It will be dangerous in some situations of IoT environment. The main reason is the lack of authentication, when the user TA communicates with the server $\mathcal{S}$ in the YPKE protocol.

3.4 The Description of Ephemeral Private Key Leakage Attack

Tian *et al.*'s original YPKE protocol was based on the HMQV protocol. However, the HMQV protocol with implicit authentication can resist ephemeral private key leakage attack. However, the adversary, who learns the value of ESK_{TA} and t_{TA} in the YPKE protocol, can use Sig_{TA} to compute the TA's private key SK_{TA1}. It is contradict to the HMQV method. Similarly, if the adversary learns the value of ESK_{TB} and t_{TB}, he also can compute TB's private key SK_{TB1}.

4 Conclusion

In this comment, we analyze the security of the YPKE protocol, and point out that the YPKE protocol still exist some weaknesses. It means that the YPKE protocol is lack of perfect forward secrecy, and cannot resist insider impersonation attack, unknown key-share attack and ephemeral private key leakage attack. In fact, the server and tags in the IoT environment have different compute capability. So it is not an easy task to design an excellent key exchange protocol in such an imbalanced network.

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] M. Chen, S. Chen, Y. Fang, "Lightweight anonymous authentication protocols for RFID systems," *IEEE/ACM Transactions on Networking*, vol. 25, no. 3, pp. 1475-1488, 2017.

[2] Y. C. Chen, W. L. Wang, M. S. Hwang, "RFID authentication protocol for anti-counterfeiting and privacy protection", in *The 9th International Conference on Advanced Communication Technology*, pp. 255–259, 2007.

[3] P. Cui, "An improved ownership transfer and mutual authentication for lightweight RFID protocols," *International Journal of Network Security*, vol. 18, no. 6, pp. 1173-1179, 2016.

[4] L. C. Huang and M. S. Hwang, "An efficient MQV key agreement scheme," *International Journal of Network Security*, vol. 16, no. 2, pp. 157–160, 2014.

[5] L. C. Huang, C. C. Lee, and M. S. Hwang, "A n2+n MQV key agreement protocol", *International Arab Journal of Information Technology*, vol. 10, no. 2, pp. 137-142, 2013.

[6] A. Juels, "Yoking-proofs for RFID tags," in *Proceedings of 2nd IEEE conference on pervasive computing and communications workshops*, pp. 138-143, 2004.

[7] H. Krawczyk, "HMQV: A high-performance secure diffie-hellman protocol," in *25th Annual International Cryptology Conference*, pp. 546-566, 2005.

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

[9] L. L. Menezes, A. Qu, M. J. Solinas, S. Vanstone, "An efficient protocol for authenticated key agreement," *Designs, Codes and Cryptography*, vol. 28, no. 2, pp. 119-134, 2003.

[10] Q. Qian, Y. Jia, R. Zhang, "A lightweight RFID security protocol based on elliptic curve cryptography," *International Journal of Network Security*, vol. 18, no. 2, pp. 354-361, 2016.

[11] C. Schnorr, "Efficient identification and signatures for smart cards," in *9th Aannual International Cryptology Conference*, pp. 239-252, 1989.

[12] Y. Tian, G. Yang, Y. Mu, "Privacy-preserving Yoking proof with key exchange in the three-party setting," *Wireless Personal Communications*, vol. 94, no. 3, pp. 1017-1034, 2017.

[13] C. H. Wei, M. S. Hwang, A. Y. H. Chin, "A mutual authentication protocol for RFID", *IEEE IT Professional*, vol. 13, no. 2, pp. 20–24, Mar. 2011.

[14] C. H. Wei, M. S. Hwang, A. Y. H. Chin, "A secure privacy and authentication protocol for passive RFID tags," *International Journal of Mobile Communications*, vol. 15, no. 3, pp. 266–277, 2017.

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.

Xinglong Zhang born in 1994, is a graduate student in the State Key Laboratory of Mathematical Engineering and Advanced Computing. His main research interests include network protocol and cyber security.