SCIENCE CHINA
Information Sciences

• LETTER •

September 2021, Vol. 64 199106:1–199106:3
https://doi.org/10.1007/s11432-019-2645-x

An enhanced key exchange protocol exhibiting key compromise impersonation attacks resistance in mobile commerce environment

Ting CHEN[1], Xinghua LI[1] & Qingfeng CHENG[2*]

[1]*School of Cyber Engineering, Xidian University, Xi'an* 710071*, China;*
[2]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou* 450001*, China*

Received 3 May 2019/Revised 6 July 2019/Accepted 9 September 2019/Published online 24 May 2021

Citation Chen T, Li X H, Cheng Q F. An enhanced key exchange protocol exhibiting key compromise impersonation attacks resistance in mobile commerce environment. Sci China Inf Sci, 2021, 64(9): 199106, https://doi.org/10.1007/s11432-019-2645-x

Dear editor,

Numerous three-party authenticated key exchange (3PAKE) protocols have been presented, which allow the establishment of a secure session by utilizing a session key shared between two clients with the assistance of a server trusted by both the clients via an unprotected network communication environment. The 3PAKE protocols can be applied to various scenarios, including mobile commerce environment. Recently, focusing on this topic, Islam et al. [1] presented an improved protocol (IAB-3PAKE) for mobile commerce environment, which remedied some flaws existing in Tan's protocol [2]. Unfortunately, the IAB-3PAKE protocol still exhibits the weakness of key compromise impersonation (KCI) attacks; an attacker can impersonate the server to share a session key with one of the clients if the client's long-term private key is acquired by the attacker. Further, with regard to the possible leakage of private keys, Wei et al. [3] reported that the private key can be recovered under some conditions.

KCI attacks are not limited to the IAB-3PAKE protocol but are also associated with the 3PAKE protocols. For instance, Yoon et al. [4] and Islam [5] designed 3PAKE protocols by focusing on biometrics and passwords, respectively, which were later observed to suffer from KCI attacks. Furthermore, considering telecare medicine information systems, key protocols have been proposed by Arshad et al. [6] and Ostad-Sharif et al. [7]. The aforementioned protocols [6, 7] have considerably contributed to this field; however, they remain vulnerable to KCI attacks. KCI security is a useful cryptographic feature and a basic security attribute for majority of the authenticated key protocols. Some two-factor/multi-factor authentication schemes also consider KCI resistance such as those reported in [8, 9]. For example, Wang et al. [8] proposed a KCI resistance method inspired by threshold password authentication. In their study, they distributed credentials utilized to authenticate users in multiple servers and reported that the KCI resistance would be valid for as long as their threshold was not exceeded.

The two-fold contribution of the present study can be given as follows.

(1) We analyze the IAB-3PAKE protocol and confirm its vulnerability to KCI attacks. After a KCI attack, the attacker can negotiate a session key with a participant whose private key was leaked to the attacker.

(2) We modify the IAB-3PAKE protocol and establish a new protocol that resolves the KCI attack issue. In the proposed protocol, we utilize the server S's public key to encrypt the messages sent to the server S and use a symmetric key shared between the client and the server to encrypt the messages sent to the client over a public channel. Finally, the proposed protocol is demonstrated to be resistant to KCI attacks.

KCI attack in the IAB-3PAKE protocol. For a detailed description of the IAB-3PAKE protocol and several preliminaries, we refer the reader to [1]. The authors presenting the IAB-3PAKE protocol reported that their method remedied the drawbacks of Tan's protocol and fulfilled the security requirements. Unfortunately, after analyzing the IAB-3PAKE protocol, we observed that KCI attacks can still occur; thus, this protocol does not satisfy the fundamental security properties associated with key authentication and key confirmation for key exchange protocols. In a KCI attack, an attacker who acquires a client's long-term private key can impersonate other protocol participants and deceive the client. In the IAB-3PAKE protocol, an attacker M who knows client A's private key d_A can impersonate the server S to deceive client A and successfully share a key with A. Similarly, if client B's private key is known to the attacker M, M can also impersonate the server S to deceive B in the same manner. More details are given below.

• A sends message (ID_A, Request) to B. M captures the message and transmits it to B. Then, B sends (ID_B, Response) to M, and M sends the message to A.

• A receives the response message and believes that

* Corresponding author (email: qingfengc2008@sina.com)

he/she is communicating with B because A is unable to detect any discrepancy in the message.

• Client A randomly selects an integer $r_A \in Z_q^*$ and computes $R_A = H(r_A \parallel d_A) \cdot Q$, $K_A = d_A \cdot U_S = d_A \cdot d_S \cdot Q$, and $C_{AS} = H(\mathrm{ID}_A \parallel \mathrm{ID}_B \parallel R_A \parallel K_A)$. Finally, A sends (ID_A, ID_B, R_A, C_{AS}) to S, which are intercepted by M.

• M forges d'_B and calculates $U'_B = d'_B \cdot Q$, where U'_B is the corresponding forged public key. After intercepting A's messages, M randomly selects a number $r'_B \in Z_q^*$. Then, M calculates $R'_B = H(r'_B \parallel d'_B) \cdot Q$ and $K'_B = d'_B \cdot U_S = d'_B \cdot d_S \cdot Q$.

• M further computes $K_A = d_A \cdot U_S = d_A \cdot d_S \cdot Q$. Then, M computes $C'_{SA} = H(\mathrm{ID}_A \parallel \mathrm{ID}_B \parallel R_A \parallel R'_B \parallel K_A)$ and sends (R'_B, C'_{SA}) to A.

• Upon receiving (R'_B, C'_{SA}) from M, A computes $H(\mathrm{ID}_A \parallel \mathrm{ID}_B \parallel R_A \parallel R'_B \| K_A)$ using his/her own R_A and K_A and the received R'_B. Evidently, the hash value $H(\mathrm{ID}_A \parallel \mathrm{ID}_B \parallel R_A \parallel R'_B \parallel K_A)$ is indistinguishable from the received C'_{SA} from A's viewpoint.

• Then, client A computes $K' = H(r_A \parallel d_A) \cdot R'_B = H(r_A \parallel d_A) \cdot H(r'_B \parallel d'_B) \cdot Q$. Subsequently, A calculates $\mathrm{SK}' = H(\mathrm{ID}_A \parallel \mathrm{ID}_B \parallel R_A \parallel R'_B \parallel K')$, where SK' is a session key shared between A and M.

Therefore, client A will believe that he/she has successfully shared a secure session key with client B through the trusted server S. Unfortunately, he/she has actually shared a session key with the attacker M, where A's long-term private key d_A has been acquired by M, who did not obtain B's long-term private key. Therefore, the protocol's authentication mechanism is compromised, exhibiting an effect similar to that exhibited by a KCI attack.

Our enhanced protocol. We propose an enhanced protocol that improves upon the IAB-3PAKE protocol and remedies this KCI attack vulnerability. The enhanced protocol utilizes the trusted server's public key and computed symmetric keys to encrypt messages transmitted among the protocol participants via a public channel to thwart potential KCI attacks.

Three parties i.e., the trusted server S, client A and client B, participate in the enhanced protocol. System parameter initialization is executed by the trusted server S, as performed for the IAB-3PAKE protocol. Furthermore, we select the advanced encryption standard as the symmetric encryption algorithm, select RSA as the public key encryption algorithm, and publish these encryption and decryption algorithms. There are three rounds in the authentication and key exchange phase. The first round R1 consists of initiator A's steps, whereas R2 contains the responder B's steps. In R3, the server S receives authentication request messages from A and B and subsequently authenticates them based on the corresponding messages. After receiving successfully authenticated messages from the server S, A and B perform the required computation and verification and finally establish a secure session key. The detailed steps of this procedure are shown below.

R1. Initiator A performs the following steps in this round.

• Randomly select two integers r_A and w_A from Z_q^* and compute $X_A = w_A \cdot Q$, $R_A = H(r_A \parallel d_A) \cdot Q$, $K_A = d_A \cdot U_S = d_A \cdot d_S \cdot Q$, and $C_{AS} = H(\mathrm{ID}_A \parallel \mathrm{ID}_B \parallel R_A \parallel K_A)$.

• Produce a public key encryption $(R_A, X_A)_{U_S}$ with the server S's public key U_S and send {ID_A, Request} and {ID_A, ID_B, C_{AS}, $(R_A, X_A)_{U_S}$} to responder B and server S, respectively. The message "Request" sent along with ID_A represents A's request for B to share a session key with him/her.

R2. Responder B executes the following steps after receiving {ID_A, Request}.

• Randomly select two integers r_B and w_B from Z_q^* and calculate $X_B = w_B \cdot Q$, $R_B = H(r_B \parallel d_B) \cdot Q$, $K_B = d_B \cdot U_S = d_B \cdot d_S \cdot Q$, and $C_{BS} = H(\mathrm{ID}_B \parallel \mathrm{ID}_A \parallel R_B \parallel K_B)$.

• Produce a public key encryption $(R_B, X_B)_{U_S}$ with server S's public key U_S and then send messages {ID_B, Response} and {ID_B, ID_A, C_{BS}, $(R_B, X_B)_{U_S}$} to A and S, respectively. The message "Response" sent along with ID_B conveys B's response of accepting the request of A.

R3. Server S performs the following steps after receiving the message {ID_A, ID_B, C_{AS}, $(R_A, X_A)_{U_S}$} from A and the message {ID_B, ID_A, C_{BS}, $(R_B, X_B)_{U_S}$} from B.

• Retrieve (R_A, X_A) and (R_B, X_B) by decrypting $(R_A, X_A)_{U_S}$ and $(R_B, X_B)_{U_S}$ using S's private key.

• Compute $K_A = d_S \cdot U_A = d_A \cdot d_S \cdot Q$, $K_B = d_S \cdot U_B = d_B \cdot d_S \cdot Q$, and $C_{AS} = H(\mathrm{ID}_A \parallel \mathrm{ID}_B \parallel R_A \parallel K_A)$ using the received R_A and the computed K_A and compute $C_{BS} = H(\mathrm{ID}_B \parallel \mathrm{ID}_A \parallel R_B \parallel K_B)$ using the received R_B and the computed K_B.

• Verify whether the computed C_{AS} = the received C_{AS}. If this equality does not hold, a message that A has not been authenticated is sent to B. Otherwise, the server S calculates $Y_{SA} = d_S \cdot X_A = w_A \cdot d_S \cdot Q$ and $C_{SA} = H(\mathrm{ID}_A \parallel \mathrm{ID}_B \parallel R_A \parallel R_B \parallel K_A)$, encrypts R_B with Y_{SA}, and sends $\{(R_B)_{Y_{SA}}, C_{SA}\}$ to A.

• Verify the consistency between the computed C_{BS} and the received C_{BS}. If these two values are not consistent, server S sends a message to A informing A that B has not been authenticated by S. Otherwise, S computes $Y_{SB} = d_S \cdot X_B = w_B \cdot d_S \cdot Q$ and $C_{SB} = H(\mathrm{ID}_B \parallel \mathrm{ID}_A \parallel R_B \parallel R_A \parallel K_B)$, encrypts R_A with Y_{SB}, and sends $\{(R_A)_{Y_{SB}}, C_{SB}\}$ to B.

• After receiving $\{(R_B)_{Y_{SA}}, C_{SA}\}$, client A calculates $Y_{SA} = w_A \cdot U_S = w_A \cdot d_S \cdot Q$, decrypts $(R_B)_{Y_{SA}}$ by utilizing Y_{SA}, and computes $C_{SA} = H(\mathrm{ID}_A \parallel \mathrm{ID}_B \parallel R_A \parallel R_B \parallel K_A)$, where R_A is A's own R_A and K_A is generated in R1. Then, A verifies that the computed C_{SA} is the received C_{SA}. If both the values are equal, A calculates $K = H(r_A \parallel d_A) \cdot R_B = H(r_A \parallel d_A) \cdot H(r_B \parallel d_B) \cdot Q$ and $SK = H(\mathrm{ID}_A \parallel \mathrm{ID}_B \parallel R_A \parallel R_B \parallel K)$.

• After B retrieves R_A encrypted by Y_{SB}, where $Y_{SB} = w_B \cdot U_S = w_B \cdot d_S \cdot Q$, B computes $C_{SB} = H(\mathrm{ID}_B \parallel \mathrm{ID}_A \parallel R_B \parallel R_A \parallel K_B)$ using B's own R_B and K_B generated in R2. Subsequently, B ensures that the computed C_{SB} is the received C_{SB}. If the result is positive, B calculates $K = H(r_B \parallel d_B) \cdot R_A = H(r_B \parallel d_B) \cdot H(r_A \parallel d_A) \cdot Q$ and $SK = H(\mathrm{ID}_A \parallel \mathrm{ID}_B \parallel R_A \parallel R_B \parallel K)$.

Consequently, A successfully shares a session key with B, and a secure communication can be subsequently established based on the session key.

The enhanced protocol resists KCI attacks. An attacker can deceive client A in interactions with other protocol entities if A's private key is known to the attacker. KCI security ensures that an attacker cannot impersonate other protocol participants to A and share a session key with A. In the enhanced protocol, an attacker is not able to impersonate other protocol parties in interactions with A. In case that client A's private key d_A is acquired by an attacker M who wishes to impersonate server S, M can intercept $\{\mathrm{ID}_A, \mathrm{ID}_B, C_{AS}, (R_A, X_A)_{U_S}\}$ and $\{(R_B)_{Y_{SA}}, C_{SA}\}$ and acquire K_A, where $K_A = d_A \cdot U_S = d_A \cdot d_S \cdot Q$. However, the encryption key Y_{SA} is only known to A and S. Because of the infeasibility of the elliptic curve discrete logarithm

problem, M cannot extract w_A and server S's private key from X_A and S's public key, respectively. Therefore, M cannot compute $Y_{SA} = d_S \cdot X_A = w_A \cdot U_S = w_A \cdot d_S \cdot Q$. If M calculates $C'_{SA} = H(\mathrm{ID}_A \parallel \mathrm{ID}_B \parallel R'_A \parallel R'_B \parallel K_A)$ with the forged R'_A and R'_B and sends C'_{SA} to A, M will fail to be authenticated because the computed C'_{SA} is not equal to the received C'_{SA}. From A's perspective, the computed $C'_{SA} = H(\mathrm{ID}_A \parallel \mathrm{ID}_B \parallel R_A \parallel R'_B \parallel K_A)$ is obtained based on A's own R_A and K_A. Thus, the enhanced protocol is resistant to KCI attacks.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. U1708262, U1736203, 61872449).

References

1 Islam S K H, Amin R, Biswas G P, et al. An improved three party authenticated key exchange protocol using hash function and elliptic curve cryptography for mobile-commerce environments. J King Saud Univ - Comput Inf Sci, 2017, 29: 311–324

2 Tan Z T. An improvement on a three-party authentication key exchange protocol using elliptic curve cryptography. J Converg Inf Tech, 2010, 5: 120–129

3 Wei W, Chen J Z, Li D, et al. Partially known information attack on SM2 key exchange protocol. Sci China Inf Sci, 2019, 62: 32105

4 Yoon E J, Yoo K Y. Robust biometric-based three-party authenticated key establishment protocols. Int J Comput Math, 2011, 88: 1144–1157

5 Islam S K H. Design and analysis of a three party password-based authenticated key exchange protocol using extended chaotic maps. Inf Sci, 2015, 312: 104–130

6 Arshad H, Rasoolzadegan A. Design of a secure authentication and key agreement scheme preserving user privacy usable in telecare medicine information systems. J Med Syst, 2016, 40: 237

7 Ostad-Sharif A, Abbasinezhad-Mood D, Nikooghadam M. A robust and efficient ECC-based mutual authentication and session key generation scheme for healthcare applications. J Med Syst, 2019, 43: 10

8 Wang D, Cheng H B, He D B, et al. On the challenges in designing identity-based privacy-preserving authentication schemes for mobile devices. IEEE Syst J, 2018, 12: 916–925

9 Liu X X, Li Y P, Qu J, et al. MAKA: provably secure multi-factor authenticated key agreement protocol. J Int Tech, 2018, 19: 669–677