A modified eCK model with stronger security for tripartite authenticated key exchange

Qingfeng Cheng*, Chuangui Ma, Fushan Wei

Zhengzhou Information Science and Technology Institute, Zhengzhou, 450002, P.R.China

Abstract

Since Bellare and Rogaway presented the first formal security model for authenticated key exchange (AKE) protocols in 1993, many formal security models have been proposed. The extended Canetti-Krawczyk (eCK) model proposed by LaMacchia et al. is currently regarded as the strongest security model for two-party AKE protocols. In this paper, we first generalize the eCK model for tripartite AKE protocols, called teCK model, and enhance the security of the new model by adding a new reveal query. In the teCK model, the adversary has stronger powers, and can learn more secret information. Then we present a new tripartite AKE protocol based on the NAXOS protocol, called T-NAXOS protocol, and analyze its security in the teCK model under the random oracle assumption.

Key words: eCK model, teCK model, authenticated key exchange, GDH assumption, random oracle assumption

1. Introduction

A tripartite authenticated key exchange (AKE) protocol allows three parties to agree upon a secret common session key over a public network. The design and analysis of secure tripartite AKE protocols have been proved to be a notorious hard problem.

In 2000, Joux [1] had proposed the first tripartite AKE protocol based on the Weil pairing. Later, Shim [2] showed that Joux's protocol suffered from the man-in-the-middle attack. Then Al-Riyami and Paterson [3] proposed a

*Corresponding author. Address: P.O.BOX 1001-7410, Zhengzhou, 450002, P.R.China
Email address: qingfengc2008@sina.com (Qingfeng Cheng)

series of tripartite AKE protocols based on the Joux's protocol. Recently, Gorantla et al. [4] have showed that Al-Riyami and Paterson's protocols can't resist key compromise impersonation (KCI) attacks. Moreover, Cheng et al. [5] also proposed two tripartite key agreement protocols from pairing in 2004, which were heuristically investigated by attempting a list of attacks. But Chien [6] showed that Cheng et al.'s protocols were vulnerable to the insider impersonation attack and the ID-based scheme even disclosed the parties's private keys. In 2005, Tso et al. [7] proposed another ID-based non-interactive tripartite AKE protocol. Unfortunately, Lim et al. [8] proved that Tso et al.'s protocol couldn't resist some impersonation attacks. Over the years, there are many tripartite AKE protocols proposed. However, the security of most of them are made through heuristic analysis. Hence, many of them still may be insecure.

Since Bellare and Rogaway [9] presented the first formal security model for AKE protocols in 1993, many formal security models have been proposed to prove the security of AKE protocols. The most famous one of these models is Canetti-Krawczyk (CK) model [10], which was proposed by Canetti and Krawczyk in 2001. But the CK model didn't cover key compromise impersonation attacks or the leakage of ephemeral private keys. So an AKE protocol proven to be secure in the CK model might still have some issues. In order to cover these attacks, LaMacchia, Lauter and Mityagin [11] proposed the extended Canetti-Krawczyk (eCK) model in 2007, which is currently regarded as the strongest security model. In recent years, two-party AKE protocols have been rigorously analyzed under various models considering different adversarial powers. However, the analysis of tripartite AKE protocols has not been as extensive as that of two-party AKE protocols. In 2004, Hitchcock et al. [12] modified the CK model [10] for tripartite AKE protocols and provided security proofs for Joux tripartite key exchange protocols [1]. But their model also had the same flaws as the original CK model.

In this paper, we propose a modification of the eCK model by adding a new reveal query to cater for tripartite AKE protocols, called teCK model. In the teCK model, the adversary may reveal any subset of $\{sk_i, esk_i, H(sk_i, esk_i),$ $sk_j, esk_j, H(sk_j, esk_j), sk_k, esk_k, H(sk_k, esk_k)\}$ on the test session according to the freshness definition, which does not contain both the ephemeral private key and static private key of one of the parties. Informally speaking, the only corruption powers that the adversary is not allowed for in the teCK model are those that would trivially break a tripartite AKE protocol. The teCK model can be used to analyze the security of tripartite AKE protocols.

Then we present a new tripartite AKE protocol based on the NAXOS protocol [11], called T-NAXOS protocol, and analyze its security in the teCK model under the random oracle assumption.

The rest of this paper is organized as follows. In Section 2, we present a formal description of the teCK model. In Section 3, we describe the T-NAXOS protocol, and analyze its security in Section 4. Finally, the conclusions will be given in Section 5.

2. Preliminaries

2.1. Assumption

In this subsection, we introduce several Diffie-Hellman problems. Let p and q be primes, where $q|p-1$. Let $G =< g >$ be a multiplicative subgroup of Z_p^*, of prime order q.

- **Computational Diffie-Hellman (CDH) Problem:** Given $U = g^u, V = g^v \in G$, where u, v were drawn at random from Z_q, compute $W = g^w \in G$, such that $CDH(U, V) = W$. That is, compute $g^w = g^{uv} \bmod p$.
- **Decisional Diffie-Hellman (DDH) Problem:** Given $U = g^u, V = g^v, W = g^w \in G$, where u, v, w were drawn at random from Z_q, determine whether $DDH(U, V, W) = 1$ or not. That is, determine whether $w = uv \bmod q$ or not.
- **Gap Diffie-Hellman (GDH) Problem:** Given $U = g^u, V = g^v \in G$, where u, v were drawn at random from Z_q, as well as an oracle that solves the DDH problem on G, compute $g^w = g^{uv} \bmod p$.

We say that G satisfies the GDH assumption if no feasible adversary can solve the GDH problem with non-negligible probability.

2.2. teCK model

In this subsection, we present the teCK model to cater for tripartite AKE protocols. The teCK model is mainly based on the eCK model. However, the adversary M has stronger powers via an additional **EphemeralHkeyReveal** query in the teCK model. The adversary M using this reveal query can learn some secret information, which is prohibited to reveal in the eCK model. For more details on the original eCK model, we refer to [11, 13, 14].

Parties. Fixing a set of n parties $\mathbf{P} = \{P_1, P_2, \cdots, P_n\}$, each of which is modeled by a probabilistic polynomial time (PPT) Turing machine, we assume that each party P_i stores a static public/private key pair (pk_i, sk_i) together with a certificate that binds the public key to that party, where pk_i is computed as g^{sk_i}. However, we do not assume that certification authority (CA) requires parties to prove possession of their static private keys.

Session. Each party P_i can be activated to execute an instance $\Pi_{i,j,k}^{sid}$ of the protocol called a session, which is identified via a session identifier $sid = (P_i, P_j, P_k, m_1, m_2, \cdots, m_l)$, where P_i is the owner of the session and P_j, P_k are the intended partners, and $m_1, m_2, \cdots, m_l$ are the list of messages that were sent and received.

Adversary Model. The adversary M is modeled as a PPT Turing machine and has full control of all communications. We assume that the adversary M is allowed to make the following queries:

- **Send**(sid, m). The adversary M sends the message m to the session sid and gets a response to the protocol specification.

- **StaticKeyReveal(**P_i**)**. The adversary M learns the static private key of the party P_i.

- **EphemeralKeyReveal**(P_i, sid). The adversary M can obtain the ephemeral private key of the party P_i, associated with the session sid.

- **EphemeralHkeyReveal**(P_i, sid). The adversary M can learn party P_i's ephemeral secret information, which is computed using the ephemeral private key and static key and associated with the session sid.

- **SessionKeyReveal**(sid). The adversary M learns a session key of a completed session sid.

- **EstablishParty**(P_i). The adversary M can arbitrarily register a legal user on behalf of the party P_i and totally control the party P_i. Parties are called honest if M does not issue this query to them.

Experiment. Initially, the adversary M is given a set $\mathbf{P}$ of honest parties. M can make any sequence of the oracle queries described above. At any time in the experiment, M selects a complete session sid owned by an honest party and makes a query **Test(**sid**)**, and is given a challenge value κ. On the test query, that is made only once during the experiment, a coin $b \in \{0,1\}$ is uniformly tossed, and the experiment answers are given $\kappa = \textbf{SessionKeyReveal}(sid)$ if $b = 1$ and a random value uniformly chosen $\kappa \in \{0,1\}^{\tau}$ if $b = 0$. M continues the experiment after the test query. At the end of the experiment, M guesses whether the challenge κ is random or not. We say the adversary M wins the experiment if the test session is freshness and he guesses the challenge correctly.

Definition 1 (Matching session). *Three sessions are said to be matching sessions, if each session has the same session identifier and has accepted the same session key.*

Definition 2 (Freshness). *Let instance $\Pi^{sid}_{i,j,k}$ be a completed session, which was executed by three honest parties P_i, P_j and P_k. We define $\Pi^{sid}_{i,j,k}$ to be fresh if none of the following five conditions hold:*

- *The adversary M reveals the session key held in $\Pi^{sid}_{i,j,k}$ or anyone of its matching sessions if the matching session exists.*

- *All matching sessions of session $\Pi^{sid}_{i,j,k}$ exist and M reveals [sk_i and esk_i], [sk_j and esk_j] or [sk_k and esk_k].*

- *one matching session of session $\Pi^{sid}_{i,j,k}$, which is associated with P_j, does not exist and M reveals [sk_i and esk_i], sk_j or [sk_k and esk_k] before the completion of session $\Pi^{sid}_{i,j,k}$.*

- *one matching session of session $\Pi^{sid}_{i,j,k}$, which is associated with P_k, does not exist and M reveals [sk_i and esk_i], [sk_j and esk_j] or sk_k before the completion of session $\Pi^{sid}_{i,j,k}$.*

- *no matching sessions of session $\Pi^{sid}_{i,j,k}$ exists and M reveals [sk_i and esk_i], sk_j or sk_k before the completion of session $\Pi^{sid}_{i,j,k}$.*

Definition 3 (teCK security). *The advantage of the adversary M with AKE protocol Π is defined as*

$$Adv^{AKE}_{\Pi}(M) = 2\Pr[M \ wins] - 1.$$

We say that the AKE protocol Π is secure in the teCK model if matching sessions compute the same session keys and no PPT adversary M has more than a negligible advantage in the above experiment.

3. T-NAXOS Protocol

Let τ be the security parameter. The AKE protocol T-NAXOS uses a group $G =< g >$ of prime order q such that the GDH assumption holds, and three hash functions $H_1 : \{0,1\}^\tau \times Z_q^* \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \rightarrow \{0,1\}^\tau$ and $H_3 : \{0,1\}^* \rightarrow \{0,1\}^\tau$, where $Z_q^* = Z_q \backslash \{0\}$, τ is a constant such that $q = O(2^\tau)$, and H_1, H_2 and H_3 are modeled as independent random oracles.

Since T-NAXOS protocol involves only three parties, we use A, B and C instead of P_i, P_j and P_k to establish a shared session key. The protocol proceeds as follows:

Step1: A, B and C choose $esk_A, esk_B, esk_B \in Z_q^*$ randomly and compute $x_1 = H_1(esk_A, sk_A), y_1 = H_1(esk_B, sk_B), z_1 = H_1(esk_C, sk_C)$ respectively. Then A, B and C send the messages as follows:

$$\begin{array}{c} A \longrightarrow B, C\text{: } X_1 = g^{esk_A + x_1} \\ B \longrightarrow A, C\text{: } Y_1 = g^{esk_B + y_1} \\ C \longrightarrow B, C\text{: } Z_1 = g^{esk_C + z_1} \end{array}$$

Step2: Upon receiving these messages, A, B and C compute the two-party session keys K_{AB}, K_{AC} and K_{BC} in the following ways:

$$\begin{array}{c} K_{AB} = H_2((pk_B Y_1)^{x_2}, Y_1^{esk_A + x_1}, A, B, X_1, Y_1) \\ K_{BC} = H_2((pk_C Z_1)^{y_2}, Z_1^{esk_B + y_1}, B, C, Y_1, Z_1) \\ K_{AC} = H_2((pk_A X_1)^{z_2}, X_1^{esk_C + z_1}, A, C, X_1, Z_1) \end{array}$$

Then A, B and C encrypt $X_2 = g^{x_2 z_2}, Y_2 = g^{x_2 y_2}$ and $Z_2 = g^{y_2 z_2}$ with the two-party session keys K_{AB}, K_{BC} and K_{AC} respectively, and send the messages X_3, Y_3 and Z_3 as follows:

$$\begin{array}{c} A \longrightarrow B\text{: } X_3 = \{X_2\}_{K_{AB}} \\ B \longrightarrow C\text{: } Y_3 = \{Y_2\}_{K_{BC}} \\ C \longrightarrow A\text{: } Z_3 = \{Z_2\}_{K_{AC}} \end{array}$$

where $x_2 = (sk_A + esk_A + H_1(esk_A, sk_A))$, $y_2 = (sk_B + esk_B + H_1(esk_B, sk_B))$, $z_2 = (sk_C + esk_C + H_1(esk_C, sk_C))$.

Step3: Upon receiving these messages, A, B and C compute the common session key K_{ABC} respectively as follows:

$$K_{ABC} = H_3(g^{x_2 y_2 z_2}, A, B, C, X_1, X_2,\ X_3, Y_1, Y_2, Y_3, Z_1, Z_2, Z_3).$$

4. Security of T-NAXOS Protocol

In this section, we will examine the T-NAXOS protocol in order to ensure that the security attributes for a tripartite AKE protocol are satisfied. We first heuristically evaluate the T-NAXOS protocol's security by attempting a list of attacks. Then we analyze its security in the teCK model under the random oracle assumption. Below is security attributes of the T-NAXOS protocol.

- **Known session key security**. It is easy to know that the session key of the T-NAXOS protocol varies with every protocol run. Since the session key is established according to the values of the parties' ephemeral private keys (esk_A, esk_B and esk_C) in that particular session, the knowledge of past session keys would not allow the adversary to deduce any future session keys.

- **Perfect forward secrecy**. Suppose that the entire static private keys (sk_A, sk_B and sk_C) have been compromised. Since the adversary does not possess ephemeral private keys employed in that particular session, the adversary is unable to derive any other previously established session keys.

- **Key compromise impersonation resilience**. Suppose that the static private key sk_A (resp., sk_B or sk_C) has been compromised and the adversary wishes to impersonate B or C in order to communicate with A. However, he is unable to forge esk_A. So he is also unable to compute the session key K_{ABC}.

- **Ephemeral key compromise impersonation resilience**. Here we suppose that the ephemeral private key esk_A has been compromised and the adversary learns (sk_B, sk_C) or (sk_B, esk_C). Since the adversary is unable to forge sk_A, he can't compute the session key K_{ABC}. In fact, even if all three ephemeral private keys are compromised, the T-NAXOS protocol is also proven secure in the teCK model.

- **Unknown key-share resilience**. If the adversary convinces a group of parties that they share some session keys with the adversary, while in fact they share the key with another party, we call the protocol suffering from unknown key-share attack. To implement such an attack on the T-NAXOS protocol, the adversary is required to learn the static private key and ephemeral private key of some entity at the same time. Otherwise, the attack hardly works. Hence, we claim that the T-NAXOS protocol has the attribute of unknown key-share resilience according to the freshness definition.

- **Key control resilience**. In the T-NAXOS protocol, no single party could force the session key to a predetermined or predicted value since the session key of the T-NAXOS protocol is derived by using the static private keys and ephemeral private keys of all three parties.

Theorem 1. *If $H_1(\cdot), H_2(\cdot)$ are two random oracles, G is a group where the GDH assumption holds, then no PPT adversary can compute the two-party session keys K_{AB}, K_{AC}, or K_{BC} in the teCK model.*

The proof of Theorem 1 is similar to the proof of Theorem 1 in [15]. The only difference between them is that only a two-party session key is proven to be secure in [15]. However, we must prove three two-party session keys to be secure in the T-NAXOS protocol according to the freshness definition for tripartite AKE. Here we omit the details.

Theorem 2. *If $H_1(\cdot), H_2(\cdot)$ and $H_3(\cdot)$ are three random oracles, encryption algorithm is secure against adaptive chosen ciphertext attack and G is a group where the GDH assumption holds, then the proposed T-NAXOS protocol is secure in the teCK model.*

From Theorem 1, we can guarantee that the adversary can't compute the two-party session keys K_{AB}, K_{AC} and K_{BC}. If the encryption algorithm used by the T-NAXOS protocol is secure against adaptive chosen ciphertext attack, the adversary will not learn X_2, Y_2 or Z_2, and can not compute the common session key K_{ABC}. It means that the proposed T-NAXOS protocol is secure in the meCK model.

5. Conclusions

We propose the teCK model by modifying the original eCK model. To our best knowledge, it is the first time to generalize the eCK model for tripartite AKE protocols. In the teCK model, the adversary has stronger powers, and even can learn all ephemeral private keys via some queries on the test session according to the freshness definition. Then we present the T-NAXOS protocol, and analyze its security in the teCK model.

Acknowledgment

This work was in part supported by the National High Technology Research and Development Program of China (No. 2009AA01Z417) and Key Scientific and Technological Project of Henan Province (No. 092101210502).

References

[1] A. Joux. A one round protocol for tripartite Diffie-Hellman. In: ANTS-IV, in: LNCS, vol.1838. Springer-Verlag, 2000, pp.385-394.

[2] K. Shim. Efficient one-round tripartite authenticated key agreement protocol from weil pairing. Electron Lett 39(2) (2003) 208-209.

[3] S.S. Al-Riyami, K.G. Paterson. Tripartite authenticated key agreement protocols from pairings. In: Cryptography and Coding 2003, in: LNCS, vol.2898. Springer-Verlag, 2003, pp.332-359.

[4] M.C. Gorantla, C. Boyd, J.M.G. Nieto. Modeling key compromise impersonation attacks on group key exchange protocols. In: PKC 2009, in: LNCS, vol.5443. Springer-Verlag, 2009, pp.105-123.

[5] Z.H. Cheng, L. Vasiu, R. Comley. Pairing-based one-round tripartite key agreement protocols. Cryptology ePrint Archive, Report 2004/079, 2004. Available from: http://eprint.iacr.org/2004/079.

[6] H.Y. Chien. Comments: insider attack on Cheng et al.'s pairing-based tripartite key agreement protocols. Cryptology ePrint Archive: Report 2005/013, 2005. Available from: http://eprint.iacr.org/2005/013.

[7] R. Tso, T. Okamoto, E. Okamoto. An id-based non-interactive tripartite key agreement protocol with k-resilience. In: Communications and Computer Networks, 2005, pp. 38-42.

[8] M.H. Lim, S. Lee, S. Moon. Cryptanalysis of Tso et al.'s id-based tripartite authenticated key agreement protocol. In: ICISS 2007, in: LNCS, vol. 4812. Springer-Verlag, 2007, pp. 64-76.

[9] M. Bellare, P. Rogaway. Entity authentication and key distribution. In: CRYPTO 1993, in: LNCS, vol.773. Springer-Verlag, 1993, pp.232-249.

[10] R. Canetti, H. Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In: EUROCRYPT 2001, in: LNCS, vol.2045. Springer-Verlag, 2001, pp.453-474.

[11] B. LaMacchia, K. Lauter, A. Mityagin. Stronger security of authenticated key exchange. In: ProvSec 2007, in: LNCS, vol.4784. Springer-Verlag, 2007, pp.1-16.

[12] Y. Hitchcock, C. Boyd, J.M.G. Nieto. Tripartite key exchange in the Canetti-Krawczyk proof model. In: INDOCRYPT 2004, in: LNCS, vol.3348. Springer-Verlag, 2004, pp.17-32.

[13] B. Ustaoğlu. Obtaining a secure and efficient key agreement protocol from (H)MQV and NAXOS, Designs, Codes and Cryptography 46(3) (2008) 329-342.

[14] H. Huang, Z.F. Cao. Strongly secure authenticated key exchange protocol based on computational Diffie-Hellman problem. Cryptology ePrint Archive, Report 2008/500, 2008. Available from: http://eprint.iacr.org/2008/500

[15] Q. Cheng, G. Han, C. Ma. A new strongly secure authenticated key exchange protocol. In: IAS 2009, 2009, pp.499-502.