Int. J. Communications, Network and System Sciences, **2012, 5, 332-336**
http://dx.doi.org/10.4236/ijcns.2012.56043 Published Online June 2012 (http://www.SciRP.org/journal/ijcns)

TUP: A New eCK-Secure AKE Protocol under the CDH Assumption

Qinglei Zhou, Zengfu Yang
School of Information and Engineering, Zhengzhou University, Zhengzhou, China
Email: ieqlzhou@zzu.edu.cn, hp4880@163.com

Received March 4, 2012; revised April 20, 2012; accepted May 14, 2012

ABSTRACT

The design and analysis of authenticated key exchange protocol is an important problem in information security area. At present, extended Canetti-Krawczyk (eCK) model provides the strongest definition of security for two party key agreement protocol, however most of the current secure protocols can not be prove to secure without Gap assumption. To avoid this phenomenon, by using twinning key technology we propose a new two party key agreement protocol TUP which is obtained by modifying the UP protocol, then in conjunction with the trapdoor test, we prove strictly that the new protocol is secure in eCK model. Compared with previous protocols, the security assumption of new proposal is more standard and weaker, and it also solves an open problem in ProvSec'09.

Keywords: Authenticated Key Exchange; Provable Security; eCK Model; Computational Diffie-Hellman Assumption; Trapdoor Test

1. Introduction

Authenticated key Exchange (AKE) protocol allows two parties to negotiate a session key through the exchange of some information in order to ensure the security of communications. Unlike key distribution protocol, the communicating parties do not need to have a pre-shared secret information and do not require a trusted third-party participation. In addition, the two sides can have the same contribution to generation of new session key. Therefore, the analysis and design of the AKE protocol become one of the important topics in the field of information security.

For the formal analysis of the AKE protocol, Canetti and Krawzcyk proposed a well-known security model denoted by CK model [1] in 2001. Recently, LaMacchia, Lauter and Mityagin Extend the former CK model (eCK [2]) that captures almost all the security properties. As it can reduce the dependence of the certificate authority (CA), relax the assumption on the ability of the attacker, maximize the information the protocol leak, the eCK model provides the strongest definition for two parties key agreement protocol and been widely recognized.

At present most of the key agreement protocols are based on the Diffie-Hellmn protocol which is more efficient and simpler than protocols that are based on digital signature or public key encryption to achieve authentication, such as MQV [3], HMQV [4] and so on. But these protocols are insecure in eCK model. In 2007, with a new ephemeral public key derivation ($X = g^{H(x,a)}$), LaMacchia, Lauter and Mityagin proposed a new protocol (NAXOS [2]) which is secure in the eCK model under the GDH assumption [5]. In 2008, According to slightly modify the NAXOS+ protocol, Lee and Park proposed NAXOS+ protocol [6] which is secure under the CDH assumption. Besides, in 2009, Ustaoglu proposed CMQV protocol [7] which is obtained from (H) MQV and NAXOS, and proof its eCK security. Then, in 2010, Lijiang Zhang modified the CMQV protocol and introduced CMQV+ [8] which has a tight security reduction.

However, according to recent studies, the NAXOS technique is vulnerable to side channel attacks and disclosure the discrete logarithm of ephemeral public keys. To avoid this attack, Ustaoglu proposed UP protocol [9] with postponed ephemeral key derivation, in which XA^d is seen as ephemeral public key, XA is seen as pseudo static key. At present, almost all the protocol use these two technologies to achieve eCK security. Like (H,C)MQV protocol which used the forking technology [10] to construct its security, In 2011, Jiaxin P and Libin W proposed TMQV [11] protocol which is secure under the CDH assumption, they also propose UP+ protocol [12] which admits higher security level with the same efficiency.

In order to remove the GDH assumption with the help of trapdoor test and postponed ephemeral key derivation (this is also an open problem proposed by Ustaoglu in ProvSec'09), we use the twin key technology [13] to ex-

tend the UP protocol and to propose TUP protocol. The new proposed protocol also has a tight security reduction. Here we denote long-term private key as (a_1, a_2), regard XA_2^d as postponed ephemeral key and XA_1 as pseudo static key. Unlike the TMQV protocol, the forking leama is not necessary for the security argument.

2. Preliminaries

2.1. Assumption

Let G denote a multiplicative cyclic group of prime order q, generated by $g \in G$. The discrete logarithm function DLOG (U) takes as input an element $U \in G$, and returns u such that $U = g^u$. The computational Diffie-Hellman function CDH (U, V) takes as input a pair or elements $U, V \in G$, and returns $X = g^{\mathrm{DLOG}(U)*\mathrm{DLOG}(V)}$. The CDH problem is defined as follows: Given $\left(g, g^u, g^v\right)$ with uniformly random choices of $u, v \in Z_q$, and compute $\mathrm{CDH}\left(g^u, g^v\right) = g^{uv}$. We say that G satisfies the CDH assumption if no feasible adversary can solve the CDH problem with non-negligilbe probability.

2.2. Trapdoor Test

Since we need to use trapdoor test [9] to analysis the new protocol TUP, we state it briefly without proof.

Trapdoor test: Let G be a cyclic group or prime order q, generated by $g \in G$, suppose U_1, r, s, Y, Z_1, Z_2 are independent random variables, where U_1, Y, Z_1, Z_2 takes values in G, and r, s, are uniformly distributed over Z_q, compute $U_2 = g^s / U_1^r$, then we have

1) U_2 is uniformly distributed over G,

2) U_1 and U_2 are independent,

3) The probability that the truth value of $Z_1^r Z_2 = Y^s$ does not agree with the truth value of

$$\left(\mathrm{CDH}\left(Y, U_1\right) = Z_1\right) \wedge \left(\mathrm{CDH}\left(Y, U_1\right) = Z_1\right)$$

is at most $1/q$.

2.3. eCK Model

In this section,we recall the eCK model [2] which will be used to analysis the new proposed protocol TUP. In this article, we use (A_1, A_2) (a_1, a_2) denote the long term public and private keys, X, x denote ephmeral public and private keys for some party A.

Parties. Each participant (denoted as A, B), which is modeled as probabilistic polynomial time (PPT) Turing Maching, can simultaneously execute multiple sessions, the session identifier (*sid*) involved in the identity of both sides and the information they exchanged. Every honest party has a secure channel to connect with certificate authority (CA) to register their public keys without any checks such as proof of possess corresponding private keys.

Adversary. An adversary M, which is also modeled as PPT turning maching, has full control over communication network between honest parties. In eCK model, we allow M to execute the following oracle queries disorderly and adaptive

1) Send (A, B, comm): Get a response from A on behalf of B, when comm is null, M can active A to execute a session with B.

2) StaticTermKeyReveal (A): Reveal the long_term key of A.

3) EphemeralKeyReveal (*sid*): Reveal the ephemeral key of session *sid*.

4) SessionKeyReveal (*sid*): Reveal the session key of sid which is completed.

5) EstablishParty (A): Register to CA with the public key of honest party A.

6) Test (*sid*): Pick i from {0, 1} randomly, if i = 0, set $k \leftarrow \{0,1\}^{\lambda}$, otherwise set $k \leftarrow K$, then return k.

Freshness. The key of session *sid* can be seen as determined by the set $\{a_1, a_2, x, b_1, b_2, y\}$, as long as M dose not execute the above query to get (a_1, a_2, x) or (b_1, b_2, y), and M does not execute SessionKeyReveal query on behalf of *sid* or its matching session sid^*, then we said *sid* is fresh.

eCK security. After execute a test query to a fresh session, M guess a number j, we say an AKE protocol is secure if the value of the following formula is negligible.

$$Adv_{\Pi}^{AKE}(M) = \left|\Pr(i = j) - 1/2\right|$$

3. Proposed AKE Protocol: TUP

In this section, According to modify the UP protocol with twin key technology, we propose a new protocol TUP, and it is secure under the CDH assumption which is more stander than the GDH assumption. **Figure 1** depicts the new protocol.

System initialization: Let $\lambda \in N$ be the security parameter and G be a cyclic group of prime order q, H_1, H_2 are hash functions modeled as random oracles where H_1 outputs integers that are half the bit-size of q. For each part A, we pick $a_1, a_2 \in Z_q^*$ randomly as its long term private key, compute its public key (A_1, A_2) and then register to CA, the party B is simulated in the same way.

Running steps: As to avoid the attack of NAXOS protocol, In the information transfer phase, when A receives the activation message, he pick $x \in Z_q^*$ randomly, compute its ephemeral public key g^x and then send it to B. After receive the message from A, B do it in the same way.

Session key derivation: to avoid the Cremers attack, we bind session key with the session state. At the end of session, both A and B can compute d, e, σ_1, σ_2, σ_3 and than the session key as described in **Figure 1**.

Brief analysis: As we can see from the derivation, compare to the UP protocol, the TUP protocol is of the same structure execept extening long-term private key to double the privious bit-size, even ignoring the σ_3, it achieves stronger security under the same environment.

In the following section, we will give its strick proof in the eCK model under the CDH assumption. Restricted by space, we simplify some parts which are same as UP protocol.

4. Security Analysis

Theorem 1. If H_1 and H_2 is modeled as random oracle, and G is a group where the CDP assumption holds, then the TUP is eCK secure AKE protocol.

Actually, let λ denote the denote the security parameter, M be a polynomially bound adversary, if M could attack TUP protocol in time at most t, involves at most n honest parties and activates at most k sessions, then we can construct a solver S which can solve the GDH problem with non-negligible probability. More precisely

$$Adv^{\mathrm{CDH}}(S) \geq \left(1/n^2k^2\right) \bullet Adv_{UP+}^{AKE}(M) - O\left(k^2/2^{\lambda}\right)$$

Proof: Since the session key of the test session is computed as $K = H(\sigma) = H(\sigma_1, \sigma_2, \sigma_3, A, B, X, Y)$ for some 7-tuple σ, the adversary M has only two ways to distinct K from random value.

1) Forging attack. At some point M queries random oracle on the same 7-tuple σ.

2) Key-replication attack. M succeeds in forcing the establishment of another session that has the same session key as the test session.

As the ephemeral public key is randomly generated, idel hash function H_2 produce no collisions (collisions happen with probability $O\left(k^2/2^{\lambda}\right)$), M must perform a forging attack to win the test game. Next we show if M can mount a successful forging attack, then we can construct a CDH solver S which uses M as a subroutine through providing M a indistinguishable simulation environment and then solve the CDH problem with non-negligible probability.

$A,(a_1,a_2)$	$B,(b_1,b_2)$
$\left(A_1=g^{a_1}, A_2=g^{a_2}\right)$	$\left(B_1=g^{b_1}, B_2=g^{b_2}\right)$
$x \leftarrow \{0,1\}^{\lambda}$	
$X=g^x$	$X \rightarrow$ $y \leftarrow \{0,1\}^{\lambda}$
	$\leftarrow Y$ $Y=g^y$
$d=H_1(A,B,X,Y)$	$e=H_1(B,A,Y,X)$
$\sigma_1=\left(YB_2^e\right)^{x+a_1}$	$\sigma_1=\left(XA_1\right)^{y+b_2e}$
$\sigma_2=\left(YB_1\right)^{x+a_2d}$	$\sigma_2=\left(XA_2^d\right)^{y+b_1}$
$\sigma_3=\left(YB_2\right)^{x+a_2}$	$\sigma_3=\left(XA_2\right)^{y+b_2}$
$\mathrm{K}=H_2(\sigma)=H_2(\sigma_1,\sigma_2,\sigma_3,\mathrm{A},\mathrm{B},\mathrm{X},\mathrm{Y})$	

Figure 1. The TUP protocol.

Given a CDH challenge (U,V), S randomly picked long-term and ephemeral private key from Z_q^* for the n paries which was involved by M and then compute the corresponding public key. Now S can answer all kinds of oracle query from M because he knows all the secret information. Then S randomly select two honest parties A, B and randomly select two sessions (denote as sid and sid^*) executed by them from all the k sessions, S will guess them with probability at least $1/n^2k^2$.

In the following step, according to the freshness principle, we will consider the following complementary events:

1) There exists session sid^* matching to the test session sid and M does not issues EphemeralKeyReveal (sid^*); and either of the following:

a) M does not issues StaticTermKeyReveal (A)—E_{1a}.

b) M does not issues EphemeralKeyReveal (sid)—E_{1b}.

2) M does not issues StaticTermKeyReveal (B), but may issues EphemeralKeyReveal (sid^*) if sid^* exists; and either of the following:

a) M does not issues StaticTermKeyReveal (A)—E_{2a}.

b) M does not issues EphemeralKeyReveal (sid)—E_{2b}.

Actually, event E_{1a} consider the case when M does not obtain (a_1, a_2, y), E_{1b} consider the case when M does not obtain (x, y), E_{2a} consider the case when M does not obtain (a_1, a_2, b_1, b_2), E_{2b} consider the case when M does not obtain (x, b_1, b_2). In any other scenario M will break freshness of the test session. For each of the complementary cases, S randomly chooses $r,s,\phi \in Z_q^*$ and modifies environment as follows:

Event. E_{1a}: S resets $A_1=U$, $A_2=g^s/U^r$, $Y=V$, according to the definition of E_{1a}, M can not distinguish with non-negligilbe probability. For each 7-tuple (σ_1, σ_2, σ_3, A, B, X, Y), M queries to H_2, S computes

$$Z_1 = f_1(\sigma_1, x, b_2) = \sigma_1\left(YB_2^e\right)^{-x} A_1^{-b_2e} = \mathrm{CDH}(A_1, Y)$$

$$Z_2 = f_1(\sigma_2, x, b_1) = \left(\sigma_2\left(YB_1\right)^{-x} A_2^{-b_1d}\right)^{d^{-1}} = \mathrm{CDH}(A_2, Y)$$

If $Z_1^r Z_2 = Y^s$ then S can answers $\mathrm{CDH}(U,V) = Z_1$. According to the trapdoor test, the probability S will fail is $1/q$ which is negligible.

Event. E_{1b}: S resets $X=U$, $Y=V$. For each 7-tuple (σ_1, σ_2, σ_3, A, B, X, Y), M queries to H_2, S computes

$$Z_1 = f_1(\sigma_1, a_1, b_2) = \sigma_1\left(YB_2^e\right)^{-a_1} X^{-b_2e} = \mathrm{CDH}(X, Y)$$

$$Z_2 = f_1(\sigma_2, a_2, b_1) = \sigma_2\left(YB_1^e\right)^{-a_1} X^{-b_1e} = \mathrm{CDH}(X, Y)$$

If $Z_1 = Z_2$ then S can answers $\mathrm{CDH}(U,V) = Z_1$.

Event. E_{2a}: S resets $A_1=V$, $A_2=V^{\phi}$, $B_2=U$, $B_1=g^s/U^r$. In the case that the maching session may not exists, as M has the ability to alter or insert messages

between honest parties, S may not known DLOG (Y). This is the reason that TUP need one more exponentiation (σ_3) than UP in the computation of shared secrets. For each 7-tuple σ, M queries to H_2, S computes

$$Z_1 = f_1(\sigma_1,\sigma_3,x) = \left(\frac{\sigma_1\left(YB_2^e\right)^{-x}}{\left(\sigma_3\left(YB_2\right)^{-x}\right)^{\phi^{-1}}}\right)^{(e-1)^{-1}}$$

$$= \mathrm{CDH}(V,B_2)$$

$$Z_2 = f_2(\sigma_2,\sigma_3,x) = \left(\frac{\left(\sigma_2\left(YB_1\right)^{-x}\right)^{d^{-1}}}{\sigma_3\left(YB_2\right)^{-x}}\right)^{\phi^{-1}}$$

$$= \mathrm{CDH}\left(V,\frac{B_1}{B_2}\right)$$

If $Z_1^r(Z_1Z_2) = V^s$ then S can answers $\mathrm{CDH}(U,V) = Z_1$.

Event. E_{2b}: S resets $X = V$, $B_2 = U$, $B_1 = g^s/U^r$, For each 7-tuple σ, M queries to H_2, S computes

$$Z_1 = f_1(\sigma_1,\sigma_3,a_1,a_2)\left(\frac{\sigma_1\left(YB_2^e\right)^{-a_1}}{\sigma_3\left(YB_2\right)^{-a_2}}\right)^{(e-1)^{-1}}$$

$$= \mathrm{CDH}(X,B_2)$$

$$Z_2 = f_2(\sigma_2,\sigma_3,a_2) = \frac{\sigma_2\left(YB_1\right)^{a_2 d}}{\sigma_3\left(YB_2\right)^{-a_2}}$$

$$= \mathrm{CDH}\left(V,\frac{B_1}{B_2}\right)$$

Similarly, If $Z_1^r(Z_1Z_2) = V^s$ then S can answers $\mathrm{CDH}(U,V) = Z_1$.

5. Protocols Comparison

In **Table 1**, we compare the new proposed protocol TUP with other previous typical protocols in some aspects. The comparison mainly focuses on the numbers of exponentiations, security assumption, tightness of security reduction and so on. For every protocol the adversary can obtain either $\tilde{x}$ as in NAXOS or x as in HMQV.

By contrast, we can see the TUP protocol has the following advantages:

1) In message exchange satage, TUP derives ephemeral key x as in MQV, so it can avoids the side channel attacks.

2) With the use of twin key technology, It can achieves security under the CDH assumption which is more stander and weaker than GDH assumption.

3) Unlike (H,C,T)MQV protocol, we did not use forking technology in security analysis, so the security reduction is more tight.

Table 1. Protocols comparison.

Protocol	Model	Efficiency	Assumption	Tight	$\tilde{x},x$
HMQV	CK01	3E	GDH	No	x
NAXOS	eCK	4E	GDH	Yes	$\tilde{x}$
NAXOS+	eCK	5E	CDH	Yes	$\tilde{x}$
CMQV	eCK	3E	GDH	No	$\tilde{x}$
CMQV+	eCK	4E	GDH	Yes	$\tilde{x}$
UP	eCK	4E	GDH	Yes	x
TMQV	eCK	5E	CDH	No	x
TUP	eCK	5E	CDH	Yes	x

Unfortunately, compared with previous protocol, the new proposed protocol has no advantages in efficiency.

6. Conclusion

In this paper, we proposed a new authenticated key exchange protocol TUP which is obtained by using twin key technology in UP protocol and added one more exponentiation in shared secrets computation. The proposed protocol also can solve an open problem in ProvSec'09. However, there is no obvious advantage regarding efficiency. So in improving the security level, to make the AKE protocol more efficient is an important goal for future work.

REFERENCES

[1] R. Canetti and H. Krawczyk, "Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels," *Lecture Notes in Computer Science*, Vol. 2045, 2001, pp. 453-474. doi:10.1007/3-540-44987-6_28

[2] B. LaMacchia, K. Lauter and A. Mityagin, "Stronger Security of Authenticated Key Exchange," *Lecture Notes in Computer Science*, Vol. 4784, 2007, pp. 1-16. doi:10.1007/978-3-540-75670-5_1

[3] L. Law, A. Menezes, M. Qu, J. Solinas and S. Vanstone, "An Efficient Protocol for Authenticated Key Agreement," *Designs, Codes and Cryptography*, Vol. 28, No. 2, 2003, pp. 119-134. doi:10.1023/A:1022595222606

[4] H. Krawczyk, "HMQV: A High-Performance Secure Diffie-Hellman Protocol," *Lecture Notes in Computer Science*, Vol. 3621, 2005, pp. 546-566. doi:10.1007/11535218_33

[5] T. Okamoto and D. Poincheval, "The Gap-Problems: A New Class of Problems for the Security of Cryptogrphic Schemes," *Lecture Notes in Computer Science*, Vol. 1992, 2001, pp. 104-118. doi:10.1007/3-540-44586-2_8

[6] J. Lee and J. H. Park, "Authenticated Key Exchange Secure under the Computational Diffe-Hellman Assumption," *Cryptology ePrint Archive*, Report 2008/344, 2008.

[7] B. Ustaoglu, "Obtaining a Secure and Efficent Key Agree-

ment Protocol from (H) MQV and NAXOS," *Designs, Codes and Crytography*, Vol. 46, No. 3, 2008, pp. 329-342. doi:10.1007/s10623-007-9159-1

[8] L. J. Zhang, "A Provably Secure Authenticated Key Exchange Protocol," *IEEE International Conference on Wireless Communications, Networking and Information Security*, Beijing, 25-27 June 2010, pp. 292-297.

[9] B. Ustaoglu, "Comparing *SessionStateReveal* and *EphemeralKeyReveal* for Diffe-Hellman Protocol," *Lecture Notes in Computer Science*, Vol. 5848, 2009, pp. 183-197. doi:10.1007/978-3-642-04642-1_16

[10] D. Pointcheval and J. Stern, "Security Arguments for Digital Signatures and Blind Signatures," *Journal of Cryptology*, Vol. 13, No. 3, 2000, pp. 361-396. doi:10.1007/s001450010003

[11] J. X. Pan and L. B. Wang, "TMQV: A Strongly eCK-Secure Diffie-Hellman Protocol without Gap Assumption," *Lecture Notes in Computer Science*, Vol. 6890, 2011, pp. 380-388. doi:10.1007/978-3-642-24316-5_27

[12] J. X. Pan, L. B. Wang and C. S. Ma, "Analysis and Improvement of an Authenticated Key Exchange Protocol," *Lecture Notes in Computer Science*, Vol. 6672, 2011, pp. 417-431. doi:10.1007/978-3-642-21031-0_31

[13] D. Cash, E. Kiltz and V. Shoup, "The Twin Diffe-Hellman Problem and Applications," " *Lecture Notes in Computer Science*, Vol. 4965, 2008, pp. 127-145. doi:10.1007/978-3-540-78967-3_8

