INFORMATICA, 2005, Vol. 16, No. 2, 275–284

© 2005 *Institute of Mathematics and Informatics, Vilnius*

An Improved Conference-Key Agreement Protocol with Forward Secrecy [*]

Yuh-Min TSENG

Department of Mathematics, National Changhua University of Education
Jin-De Campus, Chang-Hua City, Taiwan 500, R.O.C.
e-mail: tymy@ms45.hinet.net, ymtseng@cc.ncue.edu.tw

Received: July 2004

Abstract. Recently, Tzeng proposed a provably secure and fault-tolerant conference-key agreement protocol. It requires only a constant number of rounds to establish a conference key among all honest participants. This article will show that Tzeng's protocol does not offer forward secrecy. We say that a conference-key agreement protocol offers forward secrecy if the long-term secret key of any participant is compromised and will not result in the compromise of the previously established conference keys. This property is important and has been included in most key agreement protocols and standards. In this paper, an improvement based on Tzeng's protocol is proposed and it achieves forward secrecy. Under the Diffie–Hellman decision problem assumption and the random oracle model, we show that the proposed protocol can withstand passive attacks and is secure against impersonator's attacks. The improved protocol requires a constant number of rounds to compute a conference key. The improved protocol provides fault-tolerance.

Key words: forward secrecy, fault tolerant, key agreement.

1. Introduction

With the growth of network technologies, many group communication services have become the focus of future developments, such as tele-conferencing, collaborative works, and secure group communications. Efficient and secure conference-key establishment protocol design is a critical issue in providing security services for group communications over insecure channels. There are two well-known kinds of conference-key establishment protocols: conference-key distribution and conference-key agreement. In conference-key distribution protocols (Anzai *et al.*, 2001; Hwang and Yang, 1995; Tseng and Jan, 1999), there is a chairman who is responsible for generating and securely distributing a conference key to other participants involved in a conference. A conference-key agreement protocol (Bresson *et al.*, 2002; Burmester and Desmedt, 1994; Horng, 2001; Steiner *et al.*, 2000; Tzeng, 2002a) involves all participants cooperatively establishing a common key without a chairman. One advantage of a key agreement protocol over a key distribution protocol is that no participant can predetermine the common key. In the past,

[*] This research was partially supported by National Science Council, Taiwan, R.O.C., under contract no. NSC92-2213-E-018-014.

many conference-key agreement protocols have been proposed. These protocols are classified into two kinds: authenticated (Bresson *et al.*, 2002; Burmester and Desmedt, 1994; Tzeng, 2002a) and non-authenticated (Horng, 2001; Steiner *et al.*, 2000). Since a non-authenticated conference-key agreement protocol does not provide participant authentication, it is only suitable for an authenticated network channel. Generally, an authenticated conference-key agreement protocol is more flexible for various system models and requirements.

Conference-key agreement protocols are designed for various types of network connection, such as the broadcast connection, the ring connection or the tree connection, etc.. One notable result was presented by Burmester and Desmedt (1994). They proposed several key agreement protocols based on various types of network connections. Fundamentally, their non-authenticated key agreement protocol under the broadcast channel is quite efficient requiring only two rounds. However, their improved key agreement protocol with participant authentication (protocol 7) uses a sequential proof for authenticating participants. Thus, the number of rounds required in the authenticated protocol is proportional to the number of participants.

Recently, Tzeng (2002a) proposed a provably secure and fault-tolerant conference-key agreement protocol based on the broadcast channel. Fault tolerance means that even if there are many malicious participants attempting to disrupt the conference-key establishment, all other honest participants can compute the common conference key. It uses two rounds to compute a conference key. The number of rounds is independent of the number of participants. It is provably secure against passive attacks and impersonator's attacks under the variant Diffie–Hellman decision problem assumption (Boneh, 1998) and the random oracle model (Bellare and Rogaway, 1993), respectively. However, Tzeng's protocol has a weakness that it does not offer forward secrecy.

Forward secrecy means that the compromise of a long-term key(s) cannot result in the compromise of previously established conference keys. In the following, an attack scenario is described to explain the practical significance of forward secrecy. Although the probability that a malicious attacker obtains a legal participant's secret key is small, the malicious attacker could obtain any legal participant's secret key with higher probability as the time passes in the future. When the conference is significant and the conference contents must be kept secret until a special time, if the conference key is compromised and the encrypted messages will be revealed before the time is ripe, it could cause serious damage or do harm to others. Since a network or Internet is an open communication channel, eavesdroppers can intercept the transmitted messages over the open communication channel. Suppose that a malicious eavesdropper records all transmitted messages in conference-key establishment phase, while he also records all broadcast encrypted messages among participants in conference-session phase. If a conference-key establishment protocol does not provide forward secrecy, a malicious attacker could reveal the encrypted messages of a secure conference in the future. For security robustness, this property is very important and has been included in most key-agreement protocols (Bresson *et al.*, 2002; Burmester and Desmedt, 1994; Horng, 2001; Steiner *et al.*, 2000; Tseng, 2002) or standards (ANSI X9.63, 2001; IEEE Std 1363, 2000).

In this paper, we will present how Tzeng's protocol is unable to provide forward secrecy. To remedy this weakness, based on Tzeng's protocol, we propose an improvement with forward secrecy, while it remains the merits of the original protocol. The improved protocol requires a constant number of rounds to compute a conference key. The improved protocol also provides fault-tolerance.

2. Brief Review and Security Analysis of Tzeng's Protocol

In this section, we briefly review Tzeng's fault-tolerant conference-key agreement protocol (Tzeng, 2002a). Afterwards, we present that this protocol does not offer forward secrecy.

The system has the following public parameters:

- q – a large prime;
- p – a large prime such that $p = 2q + 1$;
- G_q – a subgroup of quadratic residues in Z_p^*, that is $G_q = \{i^2 | i \in Z_p^*\}$;
- g – a generator for the subgroup G_q;
- H – a one-way function from Z_q to Z_q.

Meanwhile, each user in the system has the following:

- x_i – U_i's secret key and a random value in Z_q^*;
- y_i – U_i's public key such that $y_i = g^{x_i} \bmod p$.

In this protocol, assume that there is an initiator calling for a secure conference with n participants. Without loss of generality, let $U = \{U_1, U_2, ..., U_n\}$ be the initial set of participants that want to generate a conference key. Each U_i, $1 \leqslant i \leqslant n$, knows the set U. Each U_i holds and agrees a message M to deter the replay attack (Tzeng, 2002b).

Step 1. *Secret distribution and commitment.* Initially, each participant U_i randomly selects R_i and K_i in Z_q, and $S_i \in Z_q^*$. Then, U_i constructs a polynomial $h_i(x)$ (over Z_q) with degree n that passes points $(j, (y_j^{R_i} \bmod p) \bmod q)$, $(1 \leqslant j \leqslant n)$, and $(0, K_i)$. U_i computes and broadcasts the following:

$$\begin{aligned}
&\omega_{ij} = h_i(n+j) \bmod q, \quad 1 \leqslant j \leqslant n,\\
&\alpha_i = g^{R_i} \bmod p,\\
&\gamma_i = g^{S_i} \bmod p,\\
&\delta_i = S_i^{-1}\big(H(K_i, M) - \gamma_i x_i\big) \bmod q.
\end{aligned}$$

Step 2. *Subkey computation and verification.* Upon receiving ω_{jl}, $1 \leqslant l \leqslant n$, and α_j, each participant U_i uses his secret key x_i to reconstruct the polynomial $h'_j(x)$ (over Z_q) with degree n that passes points $(n + l, \omega_{jl})$, $1 \leqslant l \leqslant n$, and $(i, (\alpha_j^{x_i} \bmod p) \bmod q)$. Let $K'_j = h'_j(0) \bmod q$. Then, U_i checks whether $g^{H(K'_j, M)} = y_j^{\gamma_j} \gamma_j^{\delta_j} \bmod p$ holds or not. If it holds, U_i broadcasts "$V_{ij} = success$". Otherwose, U_i broadcasts "$V_{ij} = failure$".

Step 3. *Fault detection.* We omit the detail description here. If any fault is detected, all honest participants must restart the protocol by deleting malicious participants from the set $U = \{U_1, U_2, \ldots, U_n\}$.

Step 4. *Conference-key computation.* We denote that the final valid participant set U' is $U' = \{U_{i1}, U_{i2}, \ldots, U_{im}\}$, where $m \leqslant n$ and the equality denotes that no fault is detected. Each participant U_i may compute the conference key

$$K = (K'_{i1} + K'_{i2} + \cdots + K'_{im}) \bmod q.$$

In the following, we present that the above protocol cannot offers forward secrecy. If a malicious user knows the secret key x_i of any participant U_i, the malicious user may compute all established conference-keys participated by U_i. Suppose that a malicious user intends to record all broadcast messages in the open network. Assume that in the past a participant U_i has ever participated in a conference. Since the malicious user knows the secret key x_i of the participant U_i, the malicious user uses the recorded messages $\omega_{jl}, 1 \leqslant l \leqslant n$, and α_j to reconstruct the polynomial $h'_j(x)(\text{over} Z_q)$ with degree n that passes points $(n + l, \omega_{jl}), 1 \leqslant l \leqslant n$, and $(i, (\alpha_j^{x_i} \bmod p) \bmod q)$, then compute$K'_j = h'_j(0) \bmod q$. Thus, he may obtain all subkeys $K'_j, 1 \leqslant j \leqslant n$, and $K = (K'_{i1} + K'_{i2} + ... + K'_{im}) \bmod q$. This is because each participant in a conference uses "ephemeral" public keys of other participants to distribute the subkey. Therefore, all previously established conference-keys participated by U_i will be disclosed if a malicious user knows the secret key x_i of participant U_i.

3. The Improved Protocol with Forward Secrecy

Here, we present an improved protocol based on Tzeng's protocol. The improved protocol with forward secrecy uses a temporary public key to distribute sub-keys, thus even the disclosure of the participant's secret key will not result in the compromise of previously established conference keys. In the improved protocol, the system model and security assumptions are the same as ones of Tzeng's protocol. Only Steps 1 and 2 in Tzeng's protocol need to be modified, and Steps 3 and 4 remain unchanged. The detailed modifications are presented as follows.

Step 1.1. *Temporary public-key distribution.* Initially, each participant randomly selects a short-term secret keys $t_i \in Z_q^*$ and an integer $v_i \in Z_q^*$, and then computes and broadcasts the following:

$$\begin{aligned} T_i &= g^{t_i} \bmod p, \\ A_i &= g^{v_i} \bmod p, \\ B_i &= v_i^{-1}\big(H(T_i, M) - A_i x_i\big) \bmod q. \end{aligned}$$

In fact, (A_i, B_i, M) can be called as a certificate for the temporary public key T_i.

Step 1.2. *Secret distribution and commitment.* Upon receiving all (T_j, A_j, B_j) ($1 \leqslant j \leqslant n$, $j \neq i$), each participant U_i checks that each T_j is really issued by U_j, i.e., checks whether $g^{H(T_j,M)} = y_j^{A_j} A_j^{B_j} \bmod p$ holds. U_i also validates whether T_j is a generator of subgroup G_q by checking $2 \leqslant w_j \leqslant p-1$ and $w_j^q \bmod p = 1$ (Menezes *et al.*, 1997). If two checks do not hold, U_i claims and broadcasts the message "U_j is faulty". If there is any participant U_j tries to send wrong T_j, the participant U_j is setted as a malicious participant and is deleted from the initial participant set U. If two checks hold, each participant U_i randomly selects R_i and K_i in Z_q, and $S_i \in Z_q^*$. Then, U_i constructs a polynomial $h_i(x)$ (over Z_q) with degree n that passes points $(j, (T_j^{R_i} \bmod p) \bmod q)$, $(1 \leqslant j \leqslant n)$, and $(0, K_i)$. U_i computes and broadcasts the following:

$$\begin{aligned}
&\omega_{ij} = h_i(n+j) \bmod q, \quad 1 \leqslant j \leqslant n,\\
&\alpha_i = g^{R_i} \bmod p,\\
&\gamma_i = g^{S_i} \bmod p,\\
&\delta_i = S_i^{-1}\bigl(H(K_i, M) - \gamma_i x_i\bigr) \bmod q.
\end{aligned}$$

Step 2. *Subkey computation and verification.* Upon receiving ω_{jl}, $1 \leqslant l \leqslant n$, and α_j, each participant U_i uses his short-term secret key t_i to reconstruct the polynomial $h'_j(x)$ (over Z_q) with degree n that passes points $(n+l, \omega_{jl})$, $1 \leqslant l \leqslant n$, and $(i, (\alpha_j^{t_i} \bmod p) \bmod q)$. Let $K'_j = h'_j(0) \bmod q$. Then, U_i checks whether $g^{H(K'_j,M)} = y_j^{\gamma_j}\gamma_j^{\delta_j} \bmod p$ holds or not. If it holds, U_i broadcasts "$V_{ij} = success$". Otherwose, U_i broadcasts "$V_{ij} = failure$".

4. Discussions

In this section, we show that the improved protocol is secure against passive attacks and impersonator attacks. Note that in the improved protocol each participant U_i increases the broadcast messages (T_i, A_i, B_i) in the Step 1.1 than Tzeng's protocol. Thus, we discuss only that the increased broadcast messages will not affect the security of Tzeng's protocol.

A. Security Against Passive Attacks

If a passive attacker is unable to obtain the established common key by eavesdropping messages transmitted over the broadcast channel, the conference-key agreement protocol is secure against passive attacks. Since an eavesdropper does not know secrets x_i and K_i of any participant U_i, the attacker's view of the messages broadcast by U_i on the broadcast channel can be simulated without knowing secrets x_i and K_i. To prove this, Tzeng's protocol adopts a variant Diffie–Hellman decision problem (Tzeng, 2002a) to show that the simulator of the attacker's view is computationally indistinguishable from the real one.

In the improved protocol, each participant increases the broadcast messages (T_i, A_i, B_i) in the Step 1.1. Therefore, we show that, the real view $(T_i, A_i, B_i, \omega_{i1}, \omega_{i2}, \ldots, \omega_{in}, \alpha_i, \gamma_i, \delta_i)$ and the attacker's view $(T_i', A_i', B_i', \omega_{i1}', \omega_{i2}', \ldots, \omega_{in}', \alpha_i', \gamma_i', \delta_i')$ on random variables $t_i' \in Z_q^*$, $v_i' \in Z_q^*$, $B_i' \in Z_q$, $\omega_{ij}' \in Z_q$ $(1 \leqslant j \leqslant n)$, $R_i' \in Z_q$, $S_i' \in Z_q^*$, $\delta_i' \in Z_q$, are computationally indistinguishable, where$T_i' = g^{t_i'} \bmod p, A_i' = g^{v_i'} \bmod p, \alpha_i' = g^{R_i'} \bmod p$ and $\gamma_i' = g^{S_i'} \bmod p$. By probability argument, since for any $A_{i0} \in G_q - \{1\}$ and $B_{i0} \in Z_q$, we have$\Pr[A_i = A_{i0}, B_i = B_{i0}] = \Pr[A_i' = A_{i0}, B_i' = B_{i0}] = \frac{1}{q(q-1)}$. For any fixed A_{i0} and B_{i0}, T_i is fixed, say T_{i0}. Thus, we need to consider two probability distributions

$$\begin{aligned}&\Pr\left[(\omega_{i1}, \omega_{i2}, \ldots, \omega_{in}, \alpha_i, \gamma_i, \delta_i) | A_i = A_{i0}, B_i = B_{i0}\right]\\&\quad = \Pr\left[(\omega_{i1}, \omega_{i2}, \ldots, \omega_{in}, \alpha_i, \gamma_i, \delta_i) | T_i = T_{i0}\right]\end{aligned}$$

and

$$\begin{aligned}&\Pr\left[(\omega_{i1}', \omega_{i2}', \ldots, \omega_{in}', \alpha_i', \gamma_i', \delta_i') | A_i' = A_{i0}, B_i' = B_{i0}\right]\\&\quad = \Pr\left[(\omega_{i1}', \omega_{i2}', \ldots, \omega_{in}', \alpha_i', \gamma_i', \delta_i')\right].\end{aligned}$$

From the above discussions, therefore we only have to consider the real view $(\omega_{i1}, \omega_{i2}, \ldots, \omega_{in}, \alpha_i, \gamma_i, \delta_i)$ and the attacker's view $(\omega_{i1}', \omega_{i2}', \ldots, \omega_{in}', \alpha_i', \gamma_i', \delta_i')$ are computationally indistinguishable. In fact, the above problem is the same as one in Tzeng's protocol, refer to (Tzeng, 2002a) for the details of the poof.

B. Security Against Impersonator's Attacks

For impersonator's adaptively chosen message attacks, it means that a malicious adversary wants to impersonate the legal participant U_i even if the broadcast channel is not authenticated. In provable security, the random oracle model (Bellare and Rogaway, 1993) is usually adopted to demonstrate the security of key establishment protocols or signature schemes. Thus, we will show the improved protocol is secure against impersonator's adaptively chosen message attacks under the random oracle model. In the improved protocol, each participant U_i broadcasts two sets of messages (T_i, A_i, B_i) in the Step 1.1 and $(\omega_{i1}, \omega_{i2}, \ldots, \omega_{in}, \alpha_i, \gamma_i, \delta_i)$ in the Step 1.2. In Tzeng's protocol, messages broadcasted by U_i are $(\omega_{i1}, \omega_{i2}, \ldots, \omega_{in}, \alpha_i, \gamma_i, \delta_i)$, which are proved "existentially un-forgeable" against the adaptively chosen message attack. Thus in the improved protocol each participant uses the short-term public key T_i to distribute the sub-key, it is needed to show that no malicious adversary is able to forge the valid T_i.

Theorem 1. *Any malicious adversary E, who can not compute the valid T_i of any legal user U_i in the random oracle model assuming that the discrete logarithm problem is intractable.*

Proof. The proof follows from that in (Pointcheval and Stern, 1996) directly. Suppose that a malicious adversary E can impersonate U_i to sign (T_i, M) with a non-negligible probability ε. Under the random oracle model, the one-way function H is a true random function, that is, $H(T_i, M)$ is an independent random variable from (T_i, M). Thus, by a probability argument, the malicious adversary E can sign the messages (T_i, M) to generate two valid signatures (A_i, B_i, h) and (A_i, B'_i, h') with a nonnegligible probability, where $A_i = g^{v_i} \bmod p, B_i = v_i^{-1}(h - A_i x_i) \bmod q, B'_i = v_i^{-1}(h' - A_i x_i) \bmod q$, and h and h' are two hash values of $H(T_i, M)$ under the random oracle model. It is easy to see that since random number v_i is used twice, the malicious adversary can compute the secret key x_i using two equations $B_i = v_i^{-1}(h - A_i x_i) \bmod q$ and $B'_i = v_i^{-1}(h' - A_i x_i) \bmod q$, which is a contradiction.

C. Forward Secrecy

In the following theorem, we show that the improved protocol provides forward secrecy under the assumption that computing the discrete logarithm problem is difficulty (Menezes *et al.*, 1997).

Theorem 2. *Under the difficulty of computing the discrete logarithm problem, the improved protocol provides forward secery.*

Proof. Without loss of generality, let $U = \{U_1, U_2, \ldots, U_n\}$ be the set of participants and they have established a conference key K at some past time τ. The conference key K is composed of all participants' sub-keys K_i $(1 \leqslant i \leqslant n)$, as $K = (K_1 + K_2 + \cdots + K_n) \bmod q$. And each K_i is distributed to other participants U_j, $(1 \leqslant j \leqslant n)$ using their short-term public keys $T_j = g^{t_j} \bmod p$. Therefore, it is necessary to obtain the short-term secret t_j. But in the following, we will show that even if an adversary with knowing x_j is unable to obtain t_j. First, the adversay tries to get t_j from $T_j = g^{t_j} \bmod p$ directly, it is clear that the adversay must face the difficulty of computing the discrete logarithm problem. Suppose that some participant U_j's secret key x_j is compromised at time $\tau + 1$, and an adversary knowing x_j tries to learn the established conference key K at time τ. Since $T_j = g^{t_j} \bmod p, A_j = g^{v_j} \bmod p$, and $B_j = v_j^{-1}(H(T_j, M) - A_j x_j) \bmod q$, the adversary may use x_j to learn v_j but not t_j, because t_j does not involved in $B_j = v_j^{-1}(H(T_j, M) - A_j x_j) \bmod q$.

On the other way, the adversary with knowing x_j tries to obtain K_j from (γ_j, δ_j), where $\gamma_j = g^{S_j} \bmod p$ and $\delta_j = S_j^{-1}(H(K_j, M) - \gamma_j x_j) \bmod q$. It is easy to see that δ_j has two other unknown variables S_j and K_j. Therefore, it is necessary to obtain the random value S_j. But the adversay tries to get S_j from $\gamma_j = g^{S_j} \bmod p$ directly, he/she also faces the difficulty of computing the discrete logarithm problem.

5. Performance Analysis

A new round is initiated each time when any number of participants needs to provide additional pieces of information to aid other participants for their key computation. Thus, a round can be a participant forwarding a piece of information to another participant or many participants simultaneously broadcasting a piece of information to others. In this sense, Tzeng's original protocol (Tzeng, 2002a) requires only one simultaneously broadcasting round for each participant. In the improved protocol, an extra broadcasting round is required. Nevertheless, the improved protocol requires only constant rounds.

Let us consider the size of messages sent by each participant in the improved protocol. Each participant U_i broadcasts two sets of messages (T_i, A_i, B_i) in the Step 1.1 and $(\omega_{i1}, \omega_{i2}, \ldots, \omega_{in}, \alpha_i, \gamma_i, \delta_i)$ in the Step 1.2, respectively. In Tzeng's protocol, messages broadcasted by U_i are $(\omega_{i1}, \omega_{i2}, \ldots, \omega_{in}, \alpha_i, \gamma_i, \delta_i)$. Therefore, the size of extra broadcast message is $2|p|+|q|$. Note that in both protocols, the size of messages that each participant sends is still proportional to the number of participants.

6. Conclusions

The improved protocol remains the merits of the original protocol with round efficiency and fault-tolerance. Under the same assumption of the variant Diffie–Hellman decision problem, we have shown that the improved protocol is secure against passive attacks. It is provably secure against impersonator attacks under the random oracle model. Furthermore, we have shown that the improved protocol offers forward secrecy, while Tzeng's protocol does not provide this property.

References

ANSI X9.63 (2001). *Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography*. ANSI.

Anzai, J., N. Matsuzaki and T. Matsumoto (2001). A flexible method for masked sharing of group keys. *IEICE Transactions on Fundamentals of Electronic Communications and Computer Science*, **E84-A**(1), 239–246.

Bellare, M., and P. Rogaway (1993). Random oracles are practical: a paradigm for designing efficient protocols. In *Proc. ACM CCS'93*. pp. 62–73.

Boneh, D. (1998). The decision Diffie–Hellman problem. In *Proc. 3rd Algorithmic Number Theorey Symp.* pp. 48–63.

Bresson, E., O. Chevassut and D. Pointcheval (2002). Dynamic group Diffie–Hellman key exchange under standard assumptions. In *Proc. Advances in Cryptology – Eurocrypt'02*. pp. 321–336.

Burmester, M., and Y. Desmedt (1994). A secure efficient conference key distribution system. In *Proc. Advances in Cryptology – Eurocrypt'94*. pp. 275–286.

Horng, G. (2001). An efficient and secure protocol for multi-party key establishment. *The Computer Journal*, **44**(5), 463–470.

Hwang, M.S., and W.P. Yang (1995). Conference key distribution schemes for secure digital mobile communications. *IEEE J. Sel. Areas Comm.*, **13**(2), 416–420.

IEEE Std 1363–2000 (2000). *Standard Specifications for Public Key Cryptography*. IEEE.

Menezes, A.J., P.C. Oorscot and S.A. Vanstone (1997). *Handbook of Applied Cryptography*. Boca Raton, FL: CRC Press.

Pointcheval, D., and J. Stern (1996). Security proofs for signature schemes. In *Proc. Advances in Cryptology – Eurocrypt'96*. pp. 387–398.

Steiner, M., G. Tsudik and M. Waidner (2000). Key agreement in dynamic peer groups. *IEEE Trans. Par. Distrib. Syst.*, **11**(8), 769–780.

Tseng, Y.M., and J.K. Jan (1999). Anonymous conference key distribution systems based on the discrete logarithm problem. *Computer Communications*, **22**(8), 749–754.

Tseng, Y.M. (2002). Robust generalized MQV key agreement protocol without using one-way hash functions. *Computer Standards & Interfaces*, **24**(3), 241–246.

Tzeng, W.G. (2002). A secure fault-tolerant conference-key agreement protocol. *IEEE Trans. on Computers*, **51**(4), 373–379.

Tzeng, W.G. (2002). Corrections to: a secure fault-tolerant conference-key agreement protocol. *IEEE Trans. on Computers*, **51**(12), 1462.

Y.-M. Tseng received the BS degree in computer science and engineering from National Chiao Tung University, Taiwan, Republic of China, in 1988; and the MS degree in computer and information engineering from National Taiwan University in 1990 and the PhD degree in applied mathematics from National Chung–Hsing University in 1999. He is currently an associate professor in the Department of Mathematics, National Changhua University of Education, Taiwan. He is a member of IEEE Communications Society and the Chinese Cryptology and Information Security Association (CCISA). His research interests include applied cryptography, communication security, network security, and mobile communications.

Modifikuotas konferencijos rakto generavimo protokolas, tenkinantis išankstinio saugumo sąlygą

Yuh-Min TSENG

Tseng'as pasiūlė konferencijos rakto generavimo protokolą. Straipsnyje parodyta, kad šis protokolas neužtikrina išankstinio saugumo (forward secrecy). Sakoma, kad konferencijos rakto generavimo protokolas tenkina išankstinio saugumo reikalavimą, kai bet kurio konferencijos dalyvio privataus rakto sukompromitavimas neturi įtakos konferencijos rakto saugumaui. Straipsnyje pasiūlytas modifikuotas Tseng'o protokolas, tenkinantis išankstinio saugumo sąlygą. Atlikta modifikuoto protokolo atsparumo įvairių atakų atžvilgiu analizė.