南　　華　　大　　學

資訊管理學系

碩士論文

一種全新以中國餘式定理為基礎的會議金鑰產生機制
A Novel CRT Based Conference-key Generation Scheme

研 究 生：徐易敬

指導教授：周志賢　博士

中華民國　一〇〇年 十二月 二十九日

南 華 大 學

資訊管理所

碩 士 學 位 論 文

一種全新以中國餘式定理為基礎的會議金鑰產生機制

A Novel CRT Based Conference-key Generation Scheme

研究生：徐易敏

經考試合格特此證明

口試委員：

許乙清

周志賢

[illegible]國[illegible]

指導教授：周志賢

系主任(所長)：吳光閔

口試日期：中華民國　100 年 12 月 29 日

誌　　　謝

經過多年的努力，終於如願的可以取得學位了，總算是完成人生其中的願望之一。這期間要感謝的人太多了，首先要感謝的當然是我的指導教授周志賢博士，這段期間，老師不嫌棄易敬是個根基基礎完成不會的學生，仍然用心仔細的指導，從零學起。再來就是感謝口試委員許乙清博士和尤國任博士，感謝您們對學生的論文無私的指教，易敬真的受益良多。

最後還是要感謝這期間無論是學長姊、學弟妹、同學和我摯愛的家人們，在求學期間的相互扶持，易敬才能努力到取得學位，除了謝謝還是謝謝，謝謝各位的幫助。

徐易敬 謹誌

一種全新以中國餘式定理為基礎的會議金鑰產生機制

學生：徐易敬　　　　　　　　　　　　指導教授：周志賢博士

南華大學　資訊管理學系碩士班

摘要

網路會議是網際網路常見的服務之一。2005 年，Ryu 等學者提出了一個網路會議密鑰分配協定。他們聲稱，他們的方法是有效率且安全的。然而，我們發現 Ryu 等人的方法，有一個安全漏洞：只要有會議參與者的公鑰和會議密鑰分配時廣播的訊息，就可以計算出該次會議之密鑰。在本研究中，我們利用中國餘式定理（Chinese Remainder Theorem），設計一種全新的會議金鑰產生機制，它可以有效地解決了安全問題。另外，本研究得到三個成果：(1) 未註冊的使用者不可能參與已認可的任何會議。(2) 未經某會議參與者的同意，一個已註冊

的使用者不得參加該會議，也不可能得到該會議的內容。（3）伺服器不能得知任何一場會議的密鑰，也不能得知該會議的參與者有那些人。在未來，我們希望利用這個方法，可以做到當參與會議人數增加時，能夠更有效率。

關鍵字： 會議金鑰分配、中國餘式定理

A Novel CRT Based Conference-key Generation Scheme

Student：I-Ching Hsu   Advisors：Dr. Jue-Sam Chou

Department of Information Management
The Graduated Program
Nan-Hua University

ABSTRACT

Network conference is one of the most popular services on the Internet. In 2005, Ryu et al. proposed a conference key distribution protocol. They claimed their scheme is efficient and secure. However, we found that their protocol has a secure loophole: without applying any private information, the conference key can be recovered only using both the broadcast message of the protocol and the public keys of the participants. In this paper, we employ the Chinese Remainder Theorem to design a novel conference key distribution scheme, which can effectively solve the security problem raised in Ryu et al.’s work. In addition, there are three outcomes of this research. First, it is impossible that an unregistered user could attend an approved conference. Second, without the agreement of all participants in a meeting, no one could obtain the conference key and the content of the conference. Third, the conference server cannot know who participates in a particular conference and cannot compute any conference key. In the future, we hope to improve our scheme which can more efficiently accommodate more users in a meeting at the same time.

Keywords: Conference key distribution, Chinese Remainder Theorem

Contents

List of Figures

List of Table

1. Introduction

With the more availability of the Internet, varied Internet services engage people’s daily life more and more. Net chat or net conference is one of the most popular Internet services. Differently from the e-mail service which is asynchronous, net conference is a synchronous service which allows many people to talk together at the same time. However, the content of the conference on an open Internet is easily eavesdropped [1] or tampered [2]. How to guard the security and the privacy of a network conference thus becomes an important issue nowadays. To solve these security problems, it urgently needs a secure conference key distribution protocol. Through the privately shared conference key, the participants in the conference can use the key to encrypt or decrypt their talking.

Reviewing from the literature, except the eavesdropping attacks and tampering attacks, there still are many different threats to the conference key distribution protocols. The research [3] shows that using uncontrolled format strings can launch mathematic-parameters attacks. Garbe [4] pointed a Distributed Denial of Service (DDoS) attack on Zhong’s work [5]. We also present an attack on Ryu eta al.’s work [6], where without any private information, one can recovery any conference key that he/she eavesdrops (The detailed attack will be shown is Section 3.). In Zhao et al.’s work [7], the authors claimed that their protocol supports multi-use property. However, we found a problem in their scheme. We assume that there are two group sets, group0 and group1. In addition, let group0 set has participates UserA, UserB and UserC , and group1 set has participates UserB, UserC and UserD. Then the malicious participant

UserB can use his/her communication session key in group1 to interfere group0 when UserB want to join into group0. Moreover, another shortcoming in Zhao et al.'s protocol is the session key exchange traffic seems too heavy. If there are n users to exchange a session key, it will cost $n \cdot (n-1) \cdot (n-1)$ times for transmitting messages. Besides, we also found that in order to protect the meeting communication toughly, some scholars like [8] apply complicated algorithms to improve the conference key security. However, these will increase more extra calculation or communication costs.

In recent studies [21-25], we found that papers [21-23] using a novel scheme to distribute the conference key. They are designed for the popular wireless and low power mobile network architecture. The author claimed that their schemes are efficient and scalable. However, their systems knows the group's conference key. This may incur unnecessary information leakage. In Konstantinou et al.'s protocols[21], their scheme requires more rounds when the number of users attending the conference increases. In Teng et al.'s protocols[22], their scheme uses bilinear pairing calculation. Although the communication rounds can be controlled in two rounds, but as more users attending the conference, the computation cost becomes inefficient. In Lu et al.'s protocols[23], when mobile node transfers encrypted message to the server; however, without the source node's identity, the server doesn't know which key should he use to decrypt the received message.

In this paper, we propose a novel conference-key agreement scheme by employing Chinese Remainder Theorem (CRT) [9]. The proposed scheme can resist from most attacks today. Compared to previous works, our approach provides a more secure and private environment to the participants of a network meeting.

The rest of the paper is organized as follows. The background knowledge such as the Chinese Remainder Theorem (CRT), distributed denial of service (DDos) attacks, and man-in-the-middle attacks are described in Section 2. Section 3 reviews Ryu et al.'s protocol and shows their weakness that we found. Section 4 presents our scheme and demonstrates an example. A security analysis is given in Section 5. Finally, we give a conclusion in Section 6.

2. Background

2.1 CRT

We first describe of the Chinese Remainder Theorem (CRT). Assume we have an unknown value x. It x divided by m_1 the remaining a_1, x divided by m_2 the remaining a_2. We can write:

$$\begin{cases} x \equiv a_1 \bmod m_1 \\ x \equiv a_2 \bmod m_2 \\ \quad\vdots \\ x \equiv a_n \bmod m_n \end{cases}$$

$$M = \prod_{i=1}^{n} m_i$$

$$M_i = M / m_i (i = 1 \text{ to } n)$$

$$x = \sum_{i=1}^{n} a_i \cdot M_i \cdot y_i \bmod M$$

$$M_i \cdot y_i = 1 \bmod m_i$$

$$x < m_1 \cdot m_2 \cdot \ldots \cdot m_n$$

For example:

There are unknowing numbers of students on the classroom. Repeatedly divided by 3, the remainder is 2; divided by 5 the remainder is 3; and divided by 7 the remainder is 2. How many students are there on the classroom? We can state the problem as follows: computes $(2\times35\times2+3\times21\times1+2\times15\times1)\bmod 105$. So the final answer is 23 students.

2.2 DDos attack

The distributed denial of service attack (DDos) was first found in 2000[10]. In general, the distributed denial of service attack by one or more co-operation to attack specific targets that it can not effectively use the network resources. And attacker usually selected high-value target specific attacks. Such as: banks, portals, websites or government credit card services unit. Mainly caused by the use of means to attack the target sites of these services does not work or provide services. Shown in Fig.1.

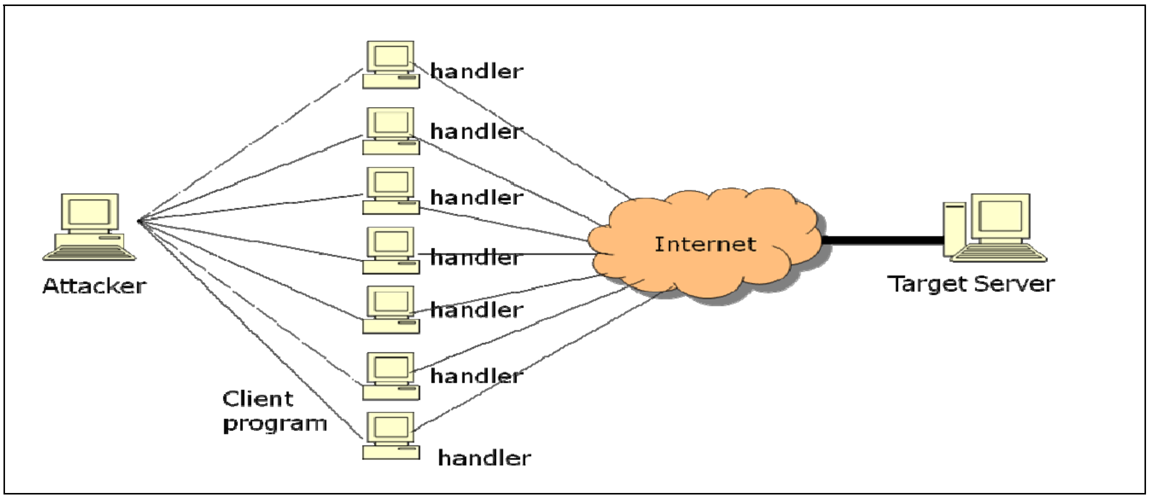


Fig. 1. DDos attack

2.3 Man in the middle attack

A malicious participant between his target and servers. Interaction with the target using the modified transmission of messages and pass each other messages with the server. Separate servers to achieve both objectives and trust, to convince them that they were about to achieve the desired results. Shown in Fig.2.

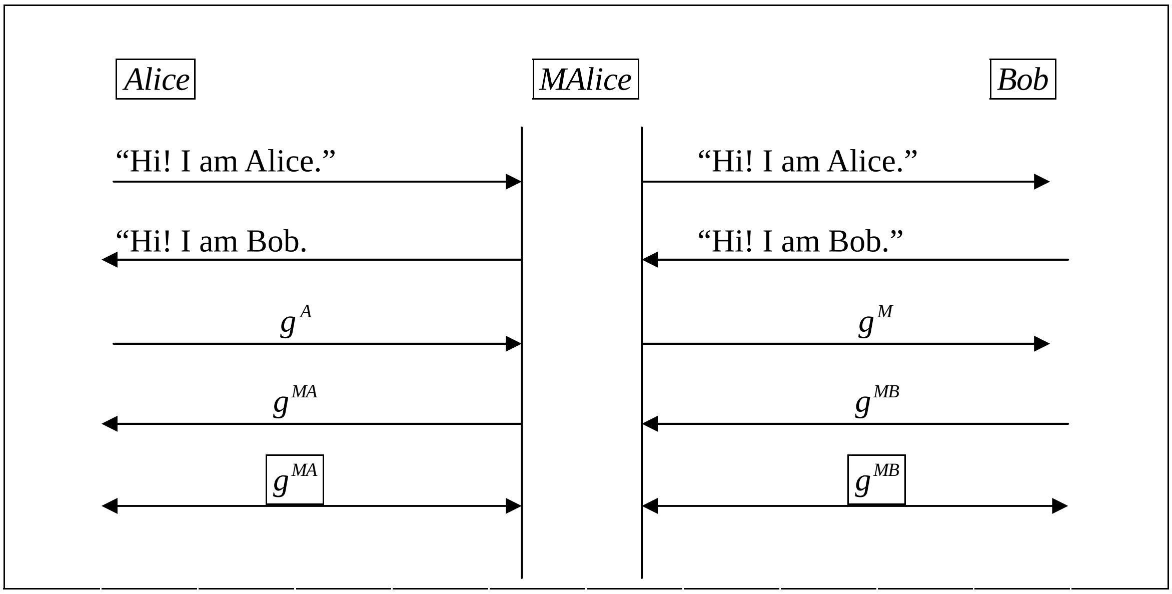


Fig. 2. Man in the middle attack

3. Eun-Kyung et al.'s protocol

3.1 Review Eun-Kyung et al.'s protocol

Eun-Kyung et al.'s conference key distribution protocol [6] is divided into three phases conference key distribution phase, conference key recovery phase and conference key verification phase. We briefly review each phase as follows:

Conference key distribution phase:

(1) Randomly choose an integer *r* and a conference key as $CK \in Z_p^*$ and set *T* as timestamp from the system and then compute $A = g^r \bmod p$, $B = r \cdot CK + H\left(T \,\|\, A\right) \cdot x_c \bmod q$.

(2) Compute the secret key shared by each U_i as

$$k_{ci} = y_i^r \bmod p,\ 1 \le i \le n.$$

(3) Construct a polynomial with degree *n* using *n* point $\left(k_{ci}, CK\right)$ as

$$P\left(x\right) = \prod_{i=1}^{n} \left(x - k_{ci}\right) + CK \bmod p = x^n + c_{n-1}x^{n-1} + ... + c_1 x + c_0 \bmod p,$$

So that $c_{n-1}, c_{n-2}, ..., c_1, c_0 \in Z_q^*.$

(4) Then broadcasts $\left\{A, B, T, c_{n-1}, c_{n-2}, ..., c_1, c_0\right\}.$

Conference key recovery phase:

On receiving $\left\{A, B, T, c_{n-1}, c_{n-2}, ..., c_1, c_0\right\}.$, each user *U_i* performs the following steps to recovery the conference key, *CK*.

(1) Check the timestamp *T*. If T is an invalid timestamp, terminate the following recovery steps.

(2) Compute the secret key shared with U_c as

$k_{ci} = A^{xi} \bmod p.$

(3) Recover the conference key CK by evaluating $P\left(k_{ci}\right)$ from the following equation system:

$$P\left(k_{ci}\right) = \left(k_{ci}\right)^n + c_{n-1}\left(k_{ci}\right)^{n-1} + \ldots + c_1 k_{ci} + c_0 \bmod p = CK \bmod p$$

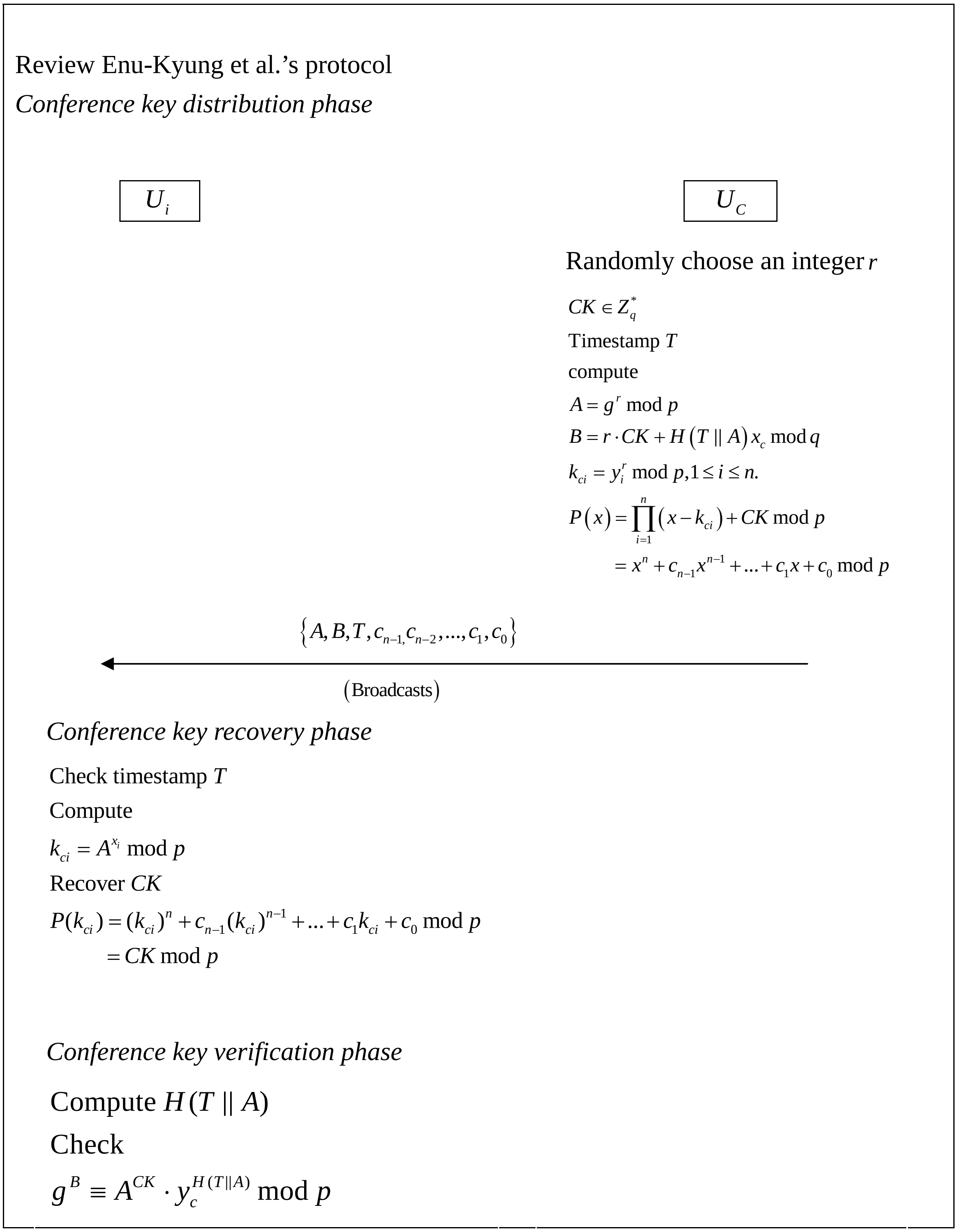
Conference key verification phase:

When obtaining CK, each uesr U_i verifies the key CK by the following equation.

Compute $H\left(T \parallel A\right)$ and check whether the following equation holds:

$$g^B \equiv A^{CK} \cdot y_c^{H(T\parallel A)} \bmod p.$$

If it is correct, each U_i insures the correctness of the distributed key. He/she can also authenticate the conference chairperson.



Review Enu-Kyung et al.'s protocol

Conference key distribution phase

U_i

U_C

Randomly choose an integer r

$CK \in Z_q^*$

Timestamp T

compute

$A = g^r \bmod p$

$B = r \cdot CK + H(T \parallel A) x_c \bmod q$

$k_{ci} = y_i^r \bmod p, 1 \le i \le n.$

$$P(x) = \prod_{i=1}^{n} (x - k_{ci}) + CK \bmod p$$
$$= x^n + c_{n-1}x^{n-1} + \ldots + c_1 x + c_0 \bmod p$$

$\{A, B, T, c_{n-1,} c_{n-2}, \ldots, c_1, c_0\}$

(Broadcasts)

Conference key recovery phase

Check timestamp T

Compute

$k_{ci} = A^{x_i} \bmod p$

Recover CK

$$P(k_{ci}) = (k_{ci})^n + c_{n-1}(k_{ci})^{n-1} + \ldots + c_1 k_{ci} + c_0 \bmod p$$
$$= CK \bmod p$$

Conference key verification phase

Compute $H(T \parallel A)$

Check

$g^B \equiv A^{CK} \cdot y_c^{H(T \parallel A)} \bmod p$

Fig. 3. Review Eun-Kyung et al.'s protocol

3.2 Weakness of Eun-Kyung Ryu et al.'s protocol

We propose an attack to compromise the confidentiality of Ryu et al.'s protocol [6] in the conference key distribution phase. As above-mentioned, U_C geneates a polynomial $P(X)$ as follows.

$$\text{where } P(X)=\prod_{i=1}^{n}(x-k_{ci})+CK \bmod p = x^n + c_{n-1}x^{n-1}+...+c_1x+c_0$$

$$\text{where } k_{ci} = A^{x_i} \bmod p$$

Here, without loss of generality, we assume that there are three participants （n=3） and their public keys are $k_{c1}=A^{x_1}, k_{c2}=A^{x_2}$ and $k_{c3}=A^{x_3}$, where x_1, x_2, and x_3 are the corresponding private keys.

Since

$$\begin{aligned}P(X)&=(x-k_{c1})(x-k_{c2})(x-k_{c3})+CK \bmod p\\ &=x^3-(k_{c1}+k_{c2}+k_{c3})x^2+(......)x-k_{c1}\times k_{c2}\times k_{c3}+CK \bmod p\end{aligned}$$

We can see that C_2 is $\left(k_{c1}+k_{c2}+k_{c3}\right)$ and C_0 is $CK-k_{c1}\times k_{c2}\times k_{c3}$. Then we can compute $k_{c1}\times k_{c2}\times k_{c3}=A^{x_1}\times A^{x_2}\times A^{x_3}=A^{x_1+x_2+x_3}=A^{c_2}$. Therefore, we can easily obtain the conference key by computing $CK=k_{c1}\times k_{c2}\times k_{c3}+C_0$. By performing above steps, we break Ryu et al's scheme.

4. Our proposed scheme

In 2009, Chang et al. proposed a OT_K^N scheme based on CRT [11], which allows a sender to transfer *N* messages without knowing which *K* messages being chosen by a receiver. (However, we found that their scheme can not satisfy the chooser's privacy [12].) Inspired by Chang et al.'s work, we apply the CRT algorithm to the conference key generation process for the network conference applications. To the best of our knowledge, we are the first to use the CRT algorithm to design a conference key generation scheme. In addition, we prove that the proposed scheme can attain the security requirements and can resist against most of attacks today.

In the next followings, we first show the used notations and then describe the three phases of the proposed scheme, namely: system initialization phase, registration phase and conference key exchange phase.

4.1 Notations

- $Server$: the trusted server;
- e, d : public/private key of Server, N is a large prime and $ed = 1 \bmod \phi(N)$.
- s : Server's secret key.
- m: the number of members in the system.
- n: the number of members which attend a meeting, n < m.
- ID_i : The identity of the participant $User_i$.
- k_i : the secret key chosen by $User_i$ and shared with $Server$.
- Mi: computed from ki and s by $Server$, and then shared with $User_i$.

- $d_1, d_2, ..., d_m$: *m* relatively large primes are secret divisors chosen by $Server$ in the initialization phase.
- C: a large number but $C < d_1 \times d_2 \times d_3 \times ... \times d_m$;
- $a_1, a_2, ..., a_m$: *m* remainders are results of *C* divided by d_i.
- ts : previously agreed time for a meeting and only known to the meeting participants.
- ks_i : one-time secret key chosen by Ueri for a conference key exchange process.
- $E_k(\bullet) / D_k(\bullet)$: encryption/decryption function pairs using the key k.
- $H(\bullet)$: Hash function
- $||$: connection symbol.

4.2 System initialization

In the time of system initialization, $Server$ generates divisors, d_1, d_1, …, d_m and a large constant *C*, where $C < d_1 \times d_2 \times d_3 \times ... \times d_m$. It also chooses a public and private key pair, (*N*, *e*), *d*, where $ed = 1 \bmod \phi(N)$. Then the $Server$ publishes its public key, (*N*, *e*), the encrypted divisors, d_1^e (mod *N*), d_2^e (mod *N*), …, d_m^e (mod *N*), and the constant, *C*, onto the public board which can be accessed by all Internet users.

4.3 Registration phase

When a user, $User_i$, wants to join the system, he/she should register to the conference server, $Server$. Then $User_i$ and $Server$ will perform the registration protocol through a secure channel as the following steps (also as shown in Fig.2). $User_i$ first selects a random number k_i as his/her secret key, and sends {ID_i, k_i} to $Server$. On receiving the message {ID_i, k_i}, $Server$ computes $M_i = H(ID_i \,||\, k_i \,||\, s)$ and stores {ID_i, k_i, M_i} into its database. Finally, $Server$ sends M_i to $User_i$. The key, M_i, will be used as an encryption key when *User*$_i$ transmits data to *Server* in the conference key exchange process.

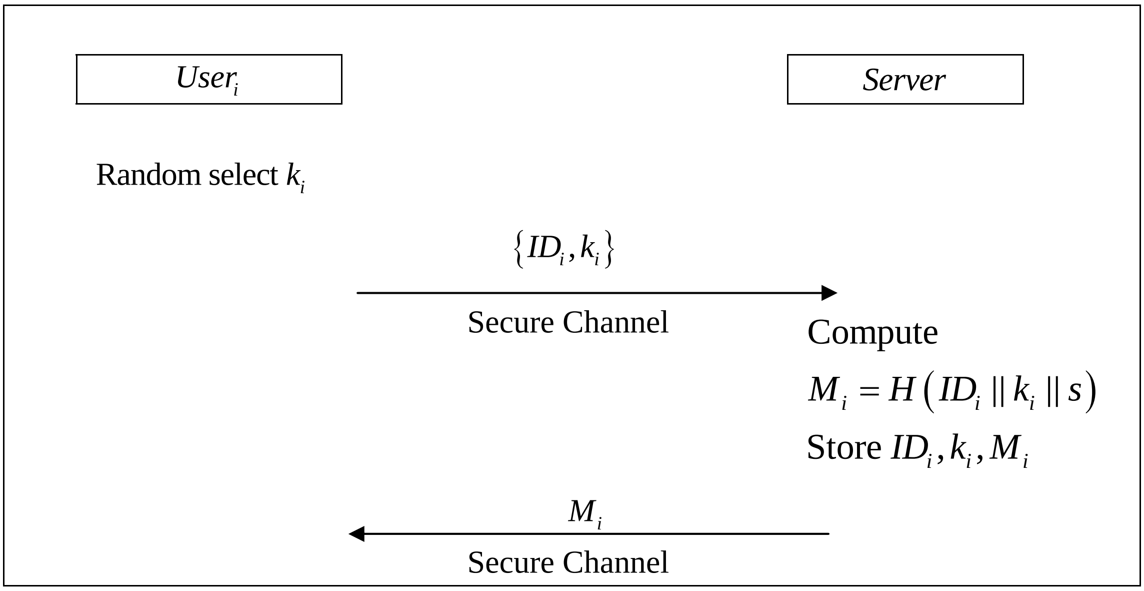


Fig. 4. Registration phase in our protocol

4.4 Conference key exchange phase

When *n* registered users want to have a meeting on the Internet, they can first agree a meeting time as *ts* and then perform the following steps to exchange a conference key. All steps are also shown in Fig.3.

- **STEP 1**

  Each user, *User*$_i$, gets C and a fresh encrypted divisor, d_i^e (mod *N*), from the public board. (Once a divisor has been used, it is removed. In addition, the *Server* will generate new fresh encrypted divisors periodically.) *User*$_i$ also randomly chooses a one-time key ks_i, and computes $r = H(ts)$, $\alpha_i = \left(r^e \cdot d_i^e\right) \bmod N$ and $\theta_i = H\left(ID_i \| \alpha_i\right)$. Then each *User*$_i$ sends $ID_i, E_{M_i}(ks_i), H(ID_i \| k_i \| ks_i), E_{ks_i}(\alpha_i), \theta_i$ to the *Server*, where k_i and M_i are shared secrecy between *User$_i$* and the *Server* .

- **STEP 2**

Without loss of the generality, we assume that the *Server* receives n times of $ID_i, E_{M_i}(ks_i), H\left(ID_i \,||\, k_i \,||\, ks_i\right), E_{ks_i}(\alpha_i), \theta_i$ from n users, where i = 1 to n, at almost the same time. Then, for each message $ID_i, E_{M_i}(ks_i), H\left(ID_i \,||\, k_i \,||\, ks_i\right), E_{ks_i}(\alpha_i), \theta_i$, the *Server* fetches the corresponding record, ID_i, k_i, M_i, from its database and performs the following processes:

(i) Decrypt $E_{M_i}(ks_i)$ to obtain *ks$_i$* by using *M$_i$* in the record.

(ii) Computs $H(ID_i \,||\, k_i \,||\, ks_i)$ to obtain $ID_i \oplus k_i$ by using the above result, *ks$_i$*.

(iii) Check if the above result, $H(ID_i \,||\, k_i \,||\, ks_i)$, is equal to $H(ID_i \,||\, k_i \,||\, ks_i)$ where *ID$_i$* and *k$_i$* are from the record. If they are not equal, the *Server* aborts this message.

(iv) Decrypt $E_{ks_i}(\alpha_i)$ by using *ks$_i$* (which is computed in (i)) to obtain α_i .

(v) Check if θ_i is equal to $H(ID_i \| \alpha_i)$ where α_i is computed in (iv). If they are equal, it implies that the message $ID_i, E_{M_i}(ks_i), H\left(ID_i \,||\, k_i \,||\, ks_i\right), E_{ks_i}(\alpha_i), \theta_i$ is a valid one from the *User$_i$.*

After confirming all *n* messages, the *Server* obtains α_i for *i* = 1 to *n*, and thus computes $\beta_i = \alpha_i^d = \left(r \cdot d_i\right) \bmod N$ for *i* = 1 to *n*. Finally, it sends the message, $\{ E_{M_i}(k_i), H(k_i \oplus ks_i), E_{ks_i}(\beta_1, \beta_2, ..., \beta_n), H(\beta_1 \,||\, \beta_2 \,||\, ... \,||\, \beta_n) \}$, to the corresponding user, *User$_i$*.

- **STEP 3**

On receiving the message { $E_{M_i}(k_i), H(k_i \oplus ks_i), E_{ks_i}(\beta_1, \beta_2, ..., \beta_n), H(\beta_1 \| \beta_2 \| ... \| \beta_n)$ }, each $User_i$ checks if $H\left(k_i \oplus ks_i\right)$ is valid. If is not, the $User_i$ rejects the message. Otherwise, the $User_i$ computes $d_i = r^{-1} \cdot \beta_i$ for *i* = 1 to *n*, and $a_1 \equiv C \bmod d_1,\ a_2 \equiv C \bmod d_2,\ a_3 \equiv C \bmod d_3, \ldots, a_n \equiv C \bmod d_n$. Finally, the $User_i$ obtains the conference key by computing $CK = \prod_{i=1}^{n} a_i \bmod N$.

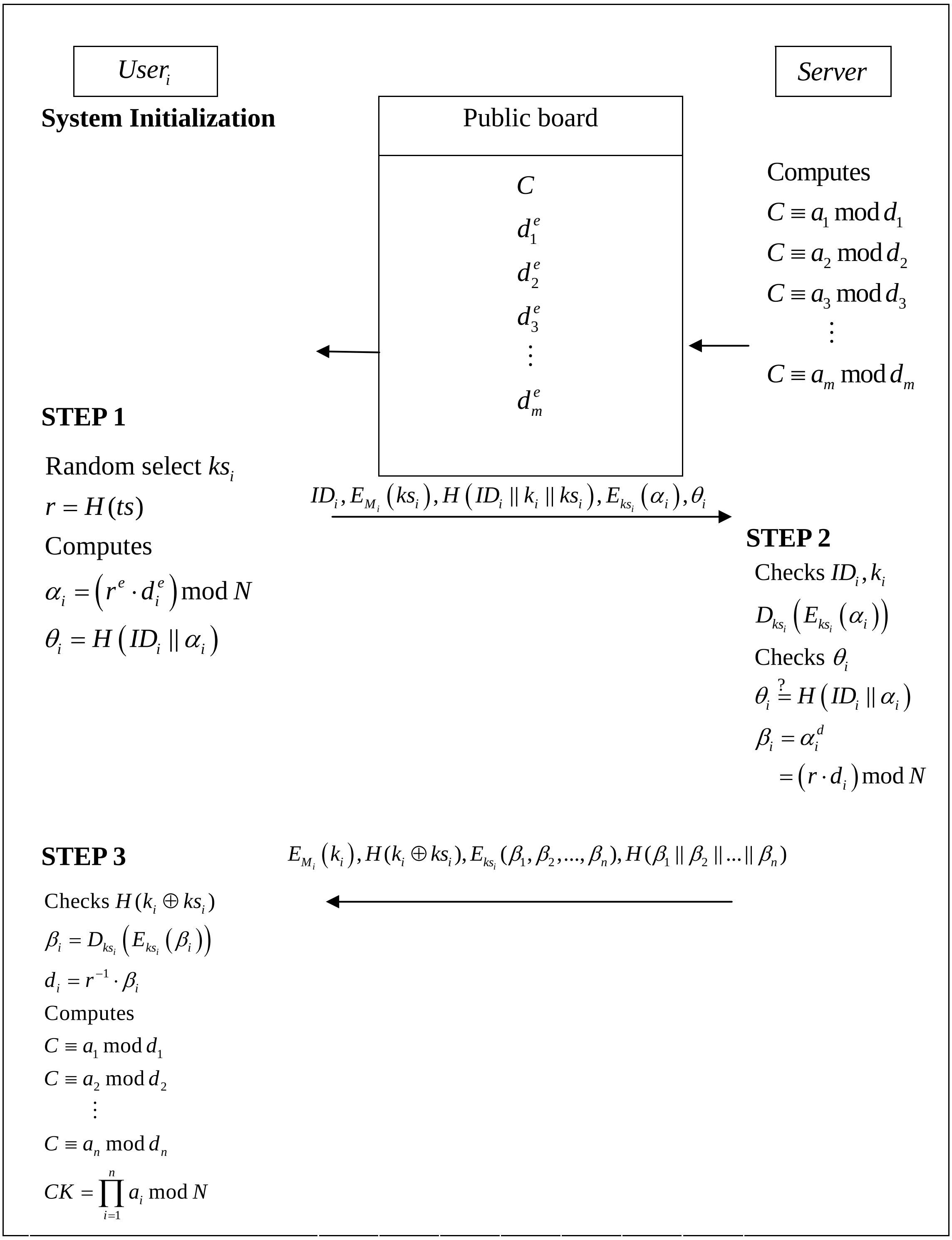

$User_i$
$Server$
System Initialization
Public board
C
d_1^e
d_2^e
d_3^e
$\vdots$
d_m^e
Computes
$C \equiv a_1 \bmod d_1$
$C \equiv a_2 \bmod d_2$
$C \equiv a_3 \bmod d_3$
$\vdots$
$C \equiv a_m \bmod d_m$
STEP 1
Random select ks_i
$r = H(ts)$
Computes
$\alpha_i = \left(r^e \cdot d_i^e\right) \bmod N$
$\theta_i = H\left(ID_i \,||\, \alpha_i\right)$
$ID_i, E_{M_i}\left(ks_i\right), H\left(ID_i \,||\, k_i \,||\, ks_i\right), E_{ks_i}\left(\alpha_i\right), \theta_i$
STEP 2
Checks ID_i, k_i
$D_{ks_i}\left(E_{ks_i}\left(\alpha_i\right)\right)$
Checks θ_i
$\theta_i \overset{?}{=} H\left(ID_i \,||\, \alpha_i\right)$
$\beta_i = \alpha_i^d$
$= \left(r \cdot d_i\right) \bmod N$
STEP 3
$E_{M_i}\left(k_i\right), H(k_i \oplus ks_i), E_{ks_i}(\beta_1, \beta_2, ..., \beta_n), H(\beta_1 \,||\, \beta_2 \,||\, ... \,||\, \beta_n)$
Checks $H(k_i \oplus ks_i)$
$\beta_i = D_{ks_i}\left(E_{ks_i}\left(\beta_i\right)\right)$
$d_i = r^{-1} \cdot \beta_i$
Computes
$C \equiv a_1 \bmod d_1$
$C \equiv a_2 \bmod d_2$
$\vdots$
$C \equiv a_n \bmod d_n$
$CK = \prod_{i=1}^{n} a_i \bmod N$

Fig. 5. Communication phase in our protocol

5. Security analysis

5.1 Security analysis

In this section, we will prove that our scheme is safe and secure. The analyses are shown in the following.

Theorem 1. Only a registered user $User_i$ can be successfully authenticated by the $Server$.

Proof:

Because $User_i$ and $Server$ can share a symmetric key M_i only through registration phase, when generating a conference key. $Server$ first uses shared M_i to decrypt the received $E_{M_i}(ks_i)$, obtaining session key ks_i. Then, and it checks if the computed $H(ID_i \| k_i \| ks_i)$ is equal to the received one if so, $Server$ believes $User_i$ is the true one as he claims.

Theorem 2. The probability of anyone rather than the conference members who can compute conference key is negligible.

Proof:

Since an adversary cannot pass the identification by the Server in the protocol, the only possible way to obtain conference key is to eavesdrop the communications between $User_i$ and $Server$.However, we show its successful probability is negligible using the following two reasons.

(i) Since $E_{ks_j}(\alpha_j)$ and $E_{ks_j}(\beta_j)$ are both protected by session key ks_j, an adversary should first obtain ks_j.

(ii) Even if the adversary obtains ks_j , he cannot extract d_j from $\alpha_j = \left(\left(r^e \cdot d_j^e\right) \bmod N = \left(r \cdot d_j\right)^e \bmod N\right)$ and $\beta_j = \left(r \cdot d_j \bmod N\right)$ because the adversary does not know N’s factoring and the group shared secrecy $r = H(ts)$. A possible remaining way is to guess conference key $CK = \prod_{i=1}^{n} a_i \bmod N$. The successful probability by guessing is $\left(1/N\right)^k$, which is negligible.

Theorem 3. Server cannot compute the conference key.

Proof:

Although the Server can obtain $\beta_j = \left(r \cdot d_j \bmod N\right)$, it does not know the group shared secrecy r and thus cannot compute the corresponding d_j and a_j. Therefore, Server cannot compute the conference key $CK = \prod_{i=1}^{n} a_i \bmod N$.

5.2 Comparison

In this section, we compare our scheme and other proposed schemes in Table 1.

	Our	[17]	[18]	[21]	[22]	[23]
Rounds	2	2	4	$\log_2 \frac{n}{3}$	2	2
Forward secrecy	Yes	Yes	Yes	Yes	Yes	Yes
Unregistered users can not join the conference	Yes	Yes	Yes	Yes	Yes	Yes
System doesn’t have the conference key	Yes	No	No	No	No	No

Tab. 1. The comparison of our scheme and other proposed schemes.

From Table.1, we can see that our study outperforms the other recent works in the aspect of system’s not knowing the conference key. This can assure the information security our scheme of communication rounds and other schemes only 2 rounds. We can do it Forward secrecy and Unregistered the users can not join the conference and other requirements. Additional, we can do that the system doesn’t have the conference key. Therefore, at user’s communication secrets, our scheme more better than other schemes.

6. Conclusion

In this paper, we review Eun-Kyung et al.'s protocol and find their protocol leaks the significant conference key. Therefore, we propose a novel CRT based conference key generation scheme using CRT to resolve the problem. After analyses, we conclude that our scheme has the following three results: (1) Only a registered user can be successfully authenticated by the server, Other people cannot participate in the conference. (2) Although the server is responsible for all conference participants to calculate the conference key for their communications. However, the server can not know who attends the conference and the content of their communications. In addition, although in the communication phase, there maybe a malicious attacker to eavesdrop on the communication content to collect relevant parameters. However, in our scheme the parameters once used will be removed in the system, prevent possible attacks. Besides, our scheme can also easily be adapted to the wireless e-commerce circumstance.

References

[1] https://www.owasp.org/index.php/Network Eavesdropping, 2009.

[2] S. Smith , S. Weingart, “Building a High-Performance, Programmable Secure Coprocessor,” *Computer Networks* 31, pp. 831–860, 1999.

[3] R. C. Seacord, ”Secure Coding in C and C++,“ Addison Wesley, September, ISBN 0-321-33572-4

[4] L. Garbe, “Denial-of-service attacks rip the internet,” *Computer Volume 33 Issue 4* pp.12-17, 2000.

[5] S. Zhong, “Efficient, Anonymous, and authenticated conference key setup in cellular wireless networks,” *Computers and Electrical Engineering 34*, pp. 357–367, 2008.

[6] E. K. Ryu, J. Y. Im, K. Y. Yoo, “Security of Tseng Jan’s conference key distribution system,” *Applied Mathematics and Computation 167*, pp. 833–839, 2005.

[7] J. Zhao, D. Gu, Y. Li ,”An efficient fault tolerant group key agreement protocol,” *Computer Communications 33*, pp.890-895, 2010.

[8] Y. Cai, X. Li, “Identity-based Conference Key Distribution Scheme Using Sealed Lock” *Seventh IEEE/ACIS International Conference on Computer and Information Science*, 2008.

[9] D. E. Knuth, “Seminumerical Algorithms,” *The Art of Computer Programming, Volume 2, ISBN 0-201-89684-2. Section 4.3.2*, pp. 286–291, exercise 4.6.2–3 ,pp. 456, 1997.

[10] G. C. Kessler, “Defenses Against Distributed Denial of Service Attacks,” *http://www.garykessler.net/library/ddos.html*, 2000.

[11] C. C. Chang, J. S. Lee, “Robust t-out-of-n oblivious transfer mechanism based on CRT,” *Journal of Network and Computer Applications 32*, pp. 226– 235, 2009.

[12] Y. Chen, J. Chou, X. Hou, “A novel k-out-of-n Oblivious Transfer Protocols Based on Bilinear Pairings,” *Cryptology ePrint Archive 027*, 2010.

[13] X. Zhao, F. Zhang, H. Tian, “Dynamic asymmetric group key agreement of ad hoc networks,” *Ad Hoc Networks 9*, pp. 928-939, 2011.

[14] B. E. Jung, S. H. Paeng, D.Y. Kim, “Attacks to Xu–Tilborg’s Conference Key

Distribution Scheme,” *Communications Letters, IEEE Vol. 8, No. 7,* pp. 446-448, 2004.

[15] K. H. Huang,Y. F. Chung, H. H. Lee, F. Lai, T. S. Chen, “A conference key agreement protocol with fault-tolerant capability,” *Computer Standards & Interfaces 31*, pp. 401–405, 2009.

[16] S. Lee, J. Kim, S. J. Hong “Security weakness of Tseng’s fault-tolerant conference-key agreement protocol,” *The Journal of Systems and Software 82,* pp. 1163–1167, 2009.

[17] Y. M. Tseng, “A communication-efficient and fault-tolerant conference-key agreement protocol with forward secrecy,” *The Journal of Systems and Software 80,* pp.1091–1101, 2007.

[18] M. H. Zheng, H. H. Zhou, J. Li, G. H. Cui, “Efficient and provably secure password-based group key agreement protocol,” *Computer Standards & Interfaces 31*, pp. 948-953, 2009.

[19] Z. You, X. Xie, “A novel group key agreement protocol for wireless mesh network,” *Computers and Electrical Engineering 37*, pp. 218-239, 2011.

[20] L. Harn, C. Lin, “Authenticated Group Key Transfer Protocol Based on Secret Sharing,” *IEEE Transactions on Computers, Vol. 59, No. 6*, pp. 842-846, 2011.

[21] E. Konstantinou, “Efficient cluster-based group key agreement protocols for wireless ad hoc networks,” *Journal of Network and Computer Applications 34* ,pp. 384-393, 2011.

[22] J. Teng and C. Wu, “A Provable Authenticated Certificateless Group Key Agreement whit Constant Rounds,” *Journal of communications and Networks Vol. 14, No. 1*, pp. 104-110, 2012.

[23] C. Lu, T. Wu, T. Shih, “Authenticated Group Key Agreement Protocol for Unbalanced Wireless Mobile Networks,” *2010 International Conference on Complex, Intelligent and Software Intensive System,* 2010.

[24] J. Nam, J. Paik, D. Won, “A security weakness in Abdalla et al.’s generic construction of a group key exchange protocol,” *Information Sciences 181*, pp. 234-238, 2011.

[25] H. Li, L. Hu, W. Yuan, H.W. Li, J. Chu, “Insider attack on a password-based group key agreement,” *Procedia Engineering 15*, pp. 1700-1704, 2011.