Dynamic Participation in a Secure Conference Scheme for Mobile Communications

Min-Shiang Hwang, *Member, IEEE*

***Abstract*— We propose a scheme to implement secure digital mobile communications. The scheme can both enable multiple users to hold a secure teleconference and also resolve the problem of allowing a participant to join dynamically or to quit a teleconference already in progress. Essentially, teleconference is a synchronous collaboration session in which participants at remote locations cooperate through wireless communications. Two requirements for the system are: privacy and authentication. Privacy signifies that an eavesdropper cannot intercept conversations of a conference. Authentication ensures that the service is not obtained fraudulently in order to avoid usage charge usage. We present a conference key distribution scheme for digital mobile communications, according to which users can share a common secret key to hold a secure teleconference over a public channel. The participants need not alter their secret information when a participant joins late or quits the conference early.**

***Index Terms*— Cryptography, mobile communications, security, teleconference.**

I. Introduction

MANY commercial products and methods have been proposed for video teleconferences through wirelined communications. However, wirelined communications can be both slow to deploy into service and expensive to install. Wireless communications, by contrast, allow people to communicate quickly and conveniently. Wireless communications have become one of the principal mediums for transmitting information.

However, wireless communications are vulnerable to relatively easy interceptions [1], such as fraudulent call attempts and intrusion or listening by third parties. Thus, the ability to prevent intrusion of various kinds is essential. For example, sensitive data must be protected against disclosure to an unauthorized person. Fraudulent modification of messages, repeating old messages or one user masquerading as another must also be prevented. Data are more vulnerable in a network than anywhere else, especially in mobile communication networks. For this reason, it is critical to develop a feasible solution to implement secure mobile communications [2]–[6].

Generally, in mobile communications each person has his own unique personal portable unit. This condition allows a caller to call a specific person, rather than calling a portable communication unit at a fixed location, at which the called party was thought to be. Thus, a chairman can hold a conference or meeting with somebody located in another area (local or remote) through wireless communications.

In conventional nonelectronic conferences, the various individuals interacting may alter as the conference proceeds. An initial group will start the conference, others join later, and some may quit early. Similar behavior can be expected to occur during a teleconference. Therefore, it is important that conference systems provide the ability to accommodate spontaneous interactions of these kinds in a conference. Specifically, these systems should allow a participant to join dynamically and to quit a secure teleconference that is already in progress. The participants in the conference need not alter their secret information when a participant joins late or leaves the conference.

We propose here a new protocol for digital mobile communications technology. This protocol enables three or more users to hold a secure conference. In general, there are four basic security objectives for mobile communications [7]:

1) privacy of conversation content during the conference;
2) privacy of information about conferees' locations during the conference;
3) prevention of fraud by ensuring that the portable units are authentic;
4) prevention of replaying attack, so that intruders are not able to obtain sensitive data by replaying a previously intercepted message.

In addition to the above basic security objectives, a conference scheme should allow dynamic joining or quitting a conference for any participant.

Since portable units must operate over long periods of time on low-power batteries, low complexity implementation of the encryption without is critical. Symmetric cryptosystems meet such criteria [13], although they require a tremendous amount of effort for key agreement [14]. Symmetric cryptosystems also require that conferees share knowledge of a secret session key, and that unauthorized users not have access to this key.

We present an effective conference key agreement scheme for digital mobile communications. Our scheme is truly dynamic in the sense that a participant joining or quitting a teleconference that is already in progress requires no modification of secret keys of any other participant in the conference. Furthermore, the computation needed to obtain the common secret session key is relatively simple, thus, our scheme can be used in low-power mobile communications.

This paper is organized as follows. In the next section, we present an effective secure conference scheme for mobile communications. We show that this scheme can achieve

Manuscript received April 15, 1997; revised August 5, 1998. This work was supported in part by the National Science Council, Taiwan, R.O.C., under Contract NSC88-2213-E-324-021.

The author is with the Department of Information Management, Chaoyang University of Technology, Wufeng, Taiwan, R.O.C.

Publisher Item Identifier S 0018-9545(99)07367-3.

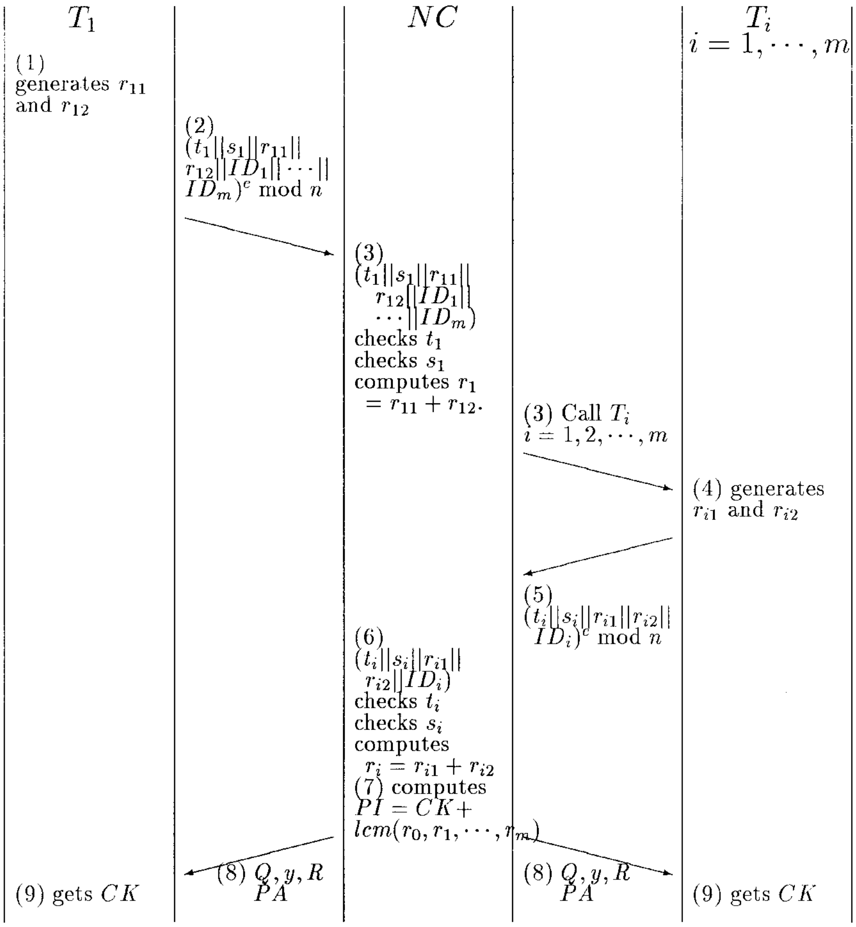
T_1
NC
T_i
$i = 1, \cdots, m$
(1) generates r_{11} and r_{12}
(2) $(t_1||s_1||r_{11}||r_{12}||ID_1||\cdots||ID_m)^e \bmod n$
(3) $(t_1||s_1||r_{11}||r_{12}||ID_1||\cdots||ID_m)$
checks t_1
checks s_1
computes $r_1 = r_{11} + r_{12}$.
(3) Call T_i $i = 1, 2, \cdots, m$
(4) generates r_{i1} and r_{i2}
(5) $(t_i||s_i||r_{i1}||r_{i2}||ID_i)^e \bmod n$
(6) $(t_i||s_i||r_{i1}||r_{i2}||ID_i)$
checks t_i
checks s_i
computes $r_i = r_{i1} + r_{i2}$
(7) computes $PI = CK + lcm(r_0, r_1, \cdots, r_m)$
(8) Q, y, R PA
(8) Q, y, R PA
(9) gets CK
(9) gets CK

Fig. 1. Conference key distribution protocol.

dynamic ability, analyze the security, and analyze the computational complexity of our scheme in Sections III-V, respectively. Section VI discusses advantages of our scheme. Finally, Section VII presents conclusions.

II. A New Scheme

According to our conference key distribution protocol (CKDP) for digital mobile communications, we assume that the network center is a trusted central authority that is responsible for key generation and key distribution, but need not retain secret keys of all conferees.

In this protocol, without loss of generality, we let personal terminal T_1 be the terminal logged in to by the chairperson U_1 who initiates the secure teleconference with m conferees, e.g., $U_1, U_2, \cdots, U_m$. We let NC be the network center, T_i be the personal terminal logged in to by U_i, ID_i be the unique identity of U_i, ID_{NC} be the unique identity of NC, t_i be the current date and time of a message to be sent, CK be a common secret session key of length 255 b of the secure teleconference chosen randomly by NC, r_{i1}, and r_{i2} be random numbers of length 256 b generated by T_i (r_{i1} and r_{i2} are added together to form r_i which is larger than CK, noneven random number of length 256 b, in that against Park's attack [8]), and || denotes concatenation. NC generates a secret number s_i for user U_i from $ID_i, s_i = f(ID_i)$, in which f is a secret one-way function that known only by the network center.

We assume that NC uses the RSA cryptosystem [9] for secure data communications. In the following protocol, T_i sends secret information using the RSA encryption under encipher key e. As e is chosen to be a small integer, T_i can perform this with low-power mobile communications. The key distribution protocol, as illustratively shown in Fig. 1, is summarized as follows.

A. Conference Key Distribution Protocol

Step 1: Initial terminal, T_1, chooses random numbers r_{11} and r_{12} such that $r_1 = r_{11} + r_{12}$ as the session key-decryption key.

Step 2: T_1 sends $(t_1||s_1||r_{11}||r_{12}||ID_i, i = 1, \cdots, m)^e$ mod n to NC.

Step 3: NC decrypts the encrypted data signal and receives $(t_1||s_1||r_{11}||r_{12}||ID_i, i = 1, \cdots, m)$. NC extracts t_1, s_1, r_{11}, r_{12}, and $ID_i, i = 1, \cdots, m$ from decrypted data. NC verifies whether $f(ID_1) = s_1$ and the validity of timestamp t_1. NC verifies T_1, computes $r_1 = r_{11} + r_{12}$, and then calls the other terminals of conferees.

Step 4: Each terminal, $T_i, i = 2, \cdots, m$, chooses random numbers r_{i1} and r_{i2} as a session key-decryption key.

Step 5: Each $T_i, i = 2, \cdots, m$ sends $(t_i \| s_i \| r_{i1} \| r_{i2} \| ID_i)^e$ mod n to NC.

Step 6: NC decrypts the encrypted data signal and receives $(t_i \| s_i \| r_{i1} \| r_{i2} \| ID_i)$. NC extracts $t_i, s_i,$ $r_{i1}, r_{i2},$ and ID_i from the decrypted data. NC verifies whether $f(ID_i) = s_i$ and the validity of timestamp t_i. NC verifies T_i and computes $r_i = r_{i1} + r_{i2}$.

Step 7: NC chooses nonzero random numbers CK and r_0, CK being a common secret session key of the secure conference, and then computes public information $PI = CK + lcm(r_0, r_1, \cdots, r_m)$ and $PA = E_{CK}(ID_{NC})$, where E is an encryption algorithm of symmetric cryptosystems [10]; lcm means a least common multiple of $(m+1)$ integers.

Step 8: NC find $Q, y,$ and R such that $PI = Q \cdot 2^y + R$, where Q and R are integers of length 256 b. NC broadcasts $Q, y, R,$ and PA to $T_i, i = 1, \cdots, m$.

Step 9: T_i obtains $CK = (Q \cdot 2^y + R)$ mod r_i and verifies whether $PA = E_{CK}(ID_{NC})$.

Any participant in the teleconference obtains the common secret session key CK in the above Step 9 as follows:

$$\begin{aligned}(Q \cdot 2^y + R) &\bmod r_i \\ &= PI \bmod r_i \\ &= (CK + lcm(r_0, r_1, \cdots, r_m)) \bmod r_i \\ &= CK. \end{aligned} \tag{1}$$

III. Dynamic Ability

True dynamic participation in the proposed scheme is achieved in the sense that the participants' secret key need not be modified in order to join the conference. When a participant joins late or quits a teleconference already in progress, the public information (PI) is only recomputed and the common secret session key (CK) of the conference is modified by the network center. All personal terminals in progress need not to alter their secret information (r_i). We explain this by the following two cases.

Case 1: When a participant U_{m+1} joins a teleconference that is already in progress, the procedures of obtaining CK for U_{m+1} are the same as in Steps 4–9 of the CKDP except that NC sends only $Q, y,$ and R, in which $Q \cdot 2^y + R = PI = CK + r_{m+1}$, to the participant.

Case 2: When a participant U_j quits a teleconference that is already in progress, the procedures of the CKDP are as follows.

Step 1: NC chooses new random numbers CK' and r'_0, and then computes new public information $PI' = CK' + lcm(r'_i, i = 0, 1, \cdots, i \neq j, \cdots, m)$ in which $r'_i = r_i + t'$ and t' is the current date and time.

Step 2: NC obtains new Q_n, y', and R_n such that $PI' = Q_n \cdot 2^{y'} + R_n$ and broadcasts (t', Q_n, y', R_n) to $T_i, i = 1, \cdots, j-1, j+1, \cdots, m$.

Step 3: T_i obtains $CK' = (Q_n \cdot 2^{y'} + R_n)$ mod $(r_i + t')$.

Despite participants joining late or quitting a conference already in progress, other participants in the conference retain their secret information r_i through the completion of the conference.

IV. Security Analysis

Both user authentication as well as session key distribution are simultaneously included in this scheme.

Because the common secret session key CK is generated randomly by the network center and hidden to public information PI with the secure random number r_i of the participants in the conference, an intruder cannot reveal CK unless he/she knows r_i or a factor of the $lcm(r_0, r_1, \cdots, r_m)$. Thus, the conferees have a secure conversation. Therefore, the proposed scheme satisfies the first security objective for mobile communications, as discussed in Section I.

To ensure that information about conferees' locations during the conference cannot be intercepted by an eavesdropper, ID_i must be protected by encryption functions. It is difficult for an intruder to obtain the location of ID_i from the equation $(t_1 \| s_1 \| r_{i1} \| r_{i2} \|\ ID_i, i = 1, \cdots, m)^e$ mod n of Steps 2 and 5 in the protocol CKDP. Therefore, the proposed scheme satisfies the second security objective for mobile communications as presented in Section I.

The network center authenticates the chairperson's identity ID_1 by verifying the correctness of ID_1 and s_1 at Step 3 of the proposed protocol. Similarly, for other users who participate in this conference invited by the chairperson, the network center can authenticate their identities ID_i and s_i by verifying the correctness of ID_i at Step 6 of the proposed protocol. This protocol ensures that a portable unit does not access the network using a false identity in order to avoid usage charge. Therefore, the proposed scheme satisfies the third security objective for mobile communications as indicated in Section I.

If an intruder obtains an intercepted message at Steps 2 or 5 of the proposed protocol, he/she is not able to complete the verification of Steps 3 and 6 in our protocol. The intruder would need to alter t_i into a new time t^* such that $(t'' - t^*) \leq \Delta t$, where t'' is the time at which the system received the illegal login message and Δt is the expected legal interval for transmission delay. Once t_i is altered, the intruder would fail verification of Steps 3 and 6 in the proposed protocols. The proposed scheme is thus secure against replaying attacks. Therefore, the proposed scheme satisfies the fourth security objective for mobile communications from Section I.

Since the protection against replay attacks relies on the synchronization of clocks, thus it may be vulnerable in the following two cases. One is that, as a result of possible communication delays or loose clock synchronization, if Δt is small legitimate authentication requests would be denied. The other area of vulnerability is of replaying attacks if Δt is large. However, many schemes involving the technology of synchronizing clocks in computer communications have been proposed in previous literature for solving the above problems [11]–[13].

When participants quit the conference at the same or different times, they know only $r_{t_0} = lcm(r_i, i = 0, 1, \cdots, i \neq j, \cdots, m)$ and $r_{t''} = lcm(r'_i, i = 0, 1, \cdots, i \neq j, \cdots, m)$. As r_{t_0} and $r_{t'}$ have no common divisor greater than the conference key, these participants who initially quit the conference cannot eavesdrop son subsequent content of the conference.

V. Computational Analysis

We now examine the complexity of the CKDP. Each personal terminal T_i needs to compute $C = (t_i \| S_i \| r_{i1} \| r_{i2} \| ID_i)^e \bmod n$ and to extract CK by computing $(Q \cdot 2^y + R) \bmod r_i$ of Steps 1, 2, 4, 5, and 9 in the protocol CKDP. The network center needs to compute m times $(C^d \bmod n)$, one time $(r_{i1} + r_{i2})$, one time $(CK + lcm(r_0, r_1, \cdots, r_m))$, and to obtain Q, y, and R of Steps 3, 6, 7, and 8 in CKDP. We examined the complexity of each step in CKDP as follows.

1) The Computational Analysis of Step 1 in CKDP: T_1 chooses two random numbers in this step. Since the computations for generating a random number are small, we ignore these computations.

2) The Computational Analysis of Step 2 in CKDP: T_1 computes $C = (t_1 \| s_1 \| r_{11} \| r_{12} \| ID_i, i = 1, \cdots, m)^e \bmod n$ in this step. Using the addition chain method [15], the complexity of computing the $(x^y \bmod z)$ form is

$$1.5l(y)[M(l(z)) + 2MOD(l(z)) + 1] \tag{2}$$

where $l(y), l(z)$ are the length of y and z, respectively, and both $M(l(z))$ and $MOD(l(z))$ denote the number of multiplications and modulus, respectively, of integers of length $l(z)$ b. The length $l(z)$ is 256 b. Since the encryption exponent e was chosen to be a small prime (i.e., $e = 3$), the length $l(y)$ is 2 b of executing Steps 2 using T_1. The number of operations is thus as follows:

$$\begin{aligned} &1.5l(y)[M(l(z)) + 2MOD(l(z)) + 1] \\ &\quad = 1.5 * 2[M(512 + 32(m+2)) \\ &\qquad + 2MOD(512 + 32(m+2)) + 1]. \end{aligned}$$

3) The Computational Analysis of Step 3 in CKDP: NC extracts t_1, s_1, r_{11}, r_{12}, and $ID_i, i = 1, \cdots, m$ from the decrypted data. In other words, NC computes $C^d \bmod n$ in this step, where C is an encrypted message of Step 2 in CKDP; d is a secret key of NC such that $ed \bmod \phi(n) = 1$; and $\phi(\cdot)$ is the Euler's totient function. Since the lengths of d and n are 256 b, the number of operations using (2) is, thus, as follows:

$$\begin{aligned} &1.5l(d)[M(l(n)) + 2MOD(l(n)) + 1] \\ &\quad = 1.5 * 256[M(512 + 32(m+2)) \\ &\qquad + 2MOD(512 + 32(m+2)) + 1]. \end{aligned}$$

4) The Computational Analysis of Step 4 in CKDP: $T_i, i = 2, 3, \cdots, m$, choose two random numbers in this step. Since the computations for generating a random number are small, we ignore these computations.

5) The Computational Analysis of Step 5 in CKDP: $T_i, i = 2, 3, \cdots, m$, compute $C = (t_1 \| s_1 \| r_{11} \| r_{12} \| ID_i, i = 1, \cdots, m)^e \bmod n$ in this step. The number of operations are the same as in Step 2 of CKDP.

6) The Computational Analysis of Step 6 in CKDP: NC extracts t_i, s_i, r_{i1}, r_{i2}, and ID_i from the decrypted data. The number of operations are the same as in Step 3 of CKDP.

7) The Computational Analysis of Step 7 in CKDP: NC computes public information $PI = CK + lcm(r_0, r_1, \cdots, r_m)$ and $PA = E_{CK}(ID_{NC})$, where E is an encryption algorithm of symmetrical cryptosystems such as DES. A software implementation of DES on an Intel 80 486/33-MHz microprocessor can perform 2.6 million bits per second [14]. Next, we examine the complexity of computing $(CK + lcm(r_0, r_1, \cdots, r_m))$, where m is the number of participants in this conference and r_i is an integer of length 256 b. The algorithm for computing $lcm(r_0, r_1, \cdots, r_m)$ is as follows. First, compute the greatest common divisor (GCD) of each pair (r_i, r_{i+1}), $i = 0, 2, 4, \cdots, (m-2)$ or $(m-1)$. We can then obtain $lcm(r_i, r_{i+1})$ by computing $r_i r_{i+1} / gcd(r_i, r_{i+1})$. Next, repeatedly compute $gcd(lcm(r_i, r_{i+1}), lcm(r_{i+2}, r_{i+3}))$ and $lcm(lcm(r_i, r_{i+1}), lcm(r_{i+2}, r_{i+3}))$ until $lcm(r_0, r_1, \cdots, r_m)$ is obtained. This algorithm requires m times GCD operations and m times divisions. Knuth has previously shown that the average number of divisions performed by Euclid's algorithm for computing GCD is approximately $0.843 \ln n + 1.47$ [15].

8) The Computational Analysis of Step 8 in CKDP: NC find Q, y, and R such that $PI = Q \cdot 2^y + R$. There are many solutions to obtain Q, y, and R such that $PI = Q \cdot 2^y + R$. We can randomly select Q, which is of length 256 b. Both of R and y are thus obtained by computing $(PI \bmod Q)$ and $\ln(PI - R/Q)/\ln 2$, respectively. If the length of R is not less than or equal to 256 b, Q is repeatedly randomly reselected until the length of R is less than or equal to 256 b. Since the length of PI is at most $256(m+1)$ b, the number of operations for computing $PI \bmod Q$ is equal to $(m+1)MOD(256)$ plus $8(m+1)$ additions, assuming that a 32-b modulo and addition facility is available. Here, $MOD(256)$ denotes the number of moduli of integers of length 256 b.

9) The Computational Analysis of Step 9 in CKDP: T_i computes $CK = (Q \cdot 2^y + R) \bmod r_i$ in this step. We analyzed the number of operations for modulus $(Q \cdot 2^y + R) \bmod r_i$. Since the length of $PI\ (= Q \cdot 2^y + R)$ is at most $256(m+1)$ b and both the lengths Q and R are 256 b, the parameter y is at most $256m$. The length of y is thus less than or equal to $\ln(256m) = 8 + \ln m$ b. To obtain $2Q$, shift Q left only one bit. To obtain $2^y \cdot Q$, shift left y bits. Therefore, to obtain $Q \cdot 2^y + R\ (= PI)$, T_i let Q shift left y bits plus R. The number of operations for computing $(Q \cdot 2^y + R) \bmod r_i$ is thus equal to $(m+1)MOD(256)$ moduli plus $(8m+9)$ additions plus y shifts, assuming that a 32-b modulo and an addition facility are available.

Since the network center is generally remotely located from the parties in the conference, the center can utilize high-powered mechanisms. Thus, the computing time can be improved significantly using parallel multiplication hardware and special high computing hardware.

However, personal terminals are movable, carried by persons or in vehicles. Since the personal terminal must operate over long periods of time on small low-power batteries, low complexity implementation is critical. The total number of operations needed by the personal terminal in our scheme is only $3M(512 + 32(m+2)) + 6MOD(512 + 32(m+2)) + (m+1)\ MOD(256) + (8m+9)\ \text{additions} + (11 + \ln m)$ shifts. Note that Steps 1, 2, and 9 in CKDP are executed by the initial terminal T_1. Steps 4, 5, and 9 in CKDP are executed by the other personal terminal $T_i, i = 2, 3, \cdots, m$, if m participants join the secure conference.

Next, we analyze the number of operations for $M(1024)$ and $MOD(1024)$. Using divide and conquer [16], together with the computational analysis of divisions, multiplications, and additions as outlined by Davida and Wells [17] is as follows:

$$D(n) = 3 * M(n) + 2S$$

$$M(n) = \begin{cases} 1, & \text{if } n = 32 \\ 3M(n/2) + 5A(n) + 2S, & \text{if } n > 32 \end{cases}$$

and

$$A(n) = \begin{cases} 1, & \text{addition if } n = 32 \\ k, & \text{additions if } n = 32 \text{ k} \end{cases}$$

where $D(n), M(n), A(n)$, and S denote the number of divisions, multiplications, additions of integers of length n bits and $S = 1$ shift, respectively. If one uses the recursion down to the 32-bit level then $M(256) = 27$ multiplications + 190 additions + 26 shifts and $M(1024) = 243$ multiplications + 2110 additions + 242 shifts.

Using divide and conquer, one can modulo two numbers as follows. Let

$$\begin{aligned} X &= a * 2^{n/2} + b \\ Y &= c * 2^{n/2} + d \\ Z &= a \bmod c \\ V &= \lfloor a/c \rfloor \\ U &= Z * 2^{n/2} + b - (V * d) \end{aligned}$$

then

$$X \bmod Y = \begin{cases} U, & \text{if } U \geq 0 \\ Y - U, & \text{otherwise} \end{cases}$$

where a, b, c, d, V, Z, and U are an integer of length $n/2$ bits; X, Y are an integer of length n bits. The number of modulo is thus as follows:

$$MOD(n) = \begin{cases} 1, & \text{if } n = 32 \\ MOD(n/2) + M(n/2) + D(n/2) & \\ \quad +A(n) + A(n/2) + S & \\ \quad = MOD(n/2) + 4M(n/2) & \\ \quad +\frac{3}{2}A(n) + 3S, & \text{if } n > 32. \end{cases}$$

If one uses the recursion down to the 32-bit level then $MOD(256) = 1$ modulus + 52 multiplications + 261 additions + 49 shifts and $MOD(1024) = 1$ modulus + 484 multiplications + 3681 additions + 479 shifts.

Finally, let $m = 14$ and ignore computing time for the shift operations. Thus, the total number of operations needed by the personal terminal in our scheme is 21 moduli + 4413 multiplications + 32 332 additions. The computing time for each operation is listed in Table I, based on using various 32-b microprocessor chips [18], [19] in our scheme. The computing time once per protocol execution is take 3.52, 1.21, and 5.58 ms when using Bellmac-32A, HP 32-b, and Inter iAPX 432 microprocessor, respectively. This is achievable in real time by a personal terminal. Thus, our schemes are a practical implementation of low-cost and low-power secure mobile communications.

TABLE I
GENERAL CHARACTERISTICS OF 32-b MICROPROCESSORS

	Bellmac-32A	HP 32-bit CPU	Intel iAPX 432
Power dissipation	0.7 Watt	4 Watts	2.5 Watts
Size of Chip	160000 MIL ^{2}	48400 MIL ^{2}	100000 MIL ^{2}
Clock frequency	10 MHz	18 MHz	8 MHz
ADD operaion	40 ns	15.3 ns	62.5 ns
Multiply operation	0.5 us	0.16 us	0.8 us
Modulus operation	0.91 us	0.29 us	1.33 us

VI. DISCUSSIONS

The concept of conference key distribution was first proposed by Ingemarsson *et al.* [20]. Subsequently, a number of studies have been carried out concerning conference key distribution systems [21]–[24]. Since these schemes are based on the discrete logarithm problem which has a high computational complexity (modular exponentiation) as the fundamental arithmetic, they are unsuitable to use in inexpensive low-power mobile communication systems [3], [4], [8]. Recently, Hwang and Yang proposed two efficient conference key distribution schemes for mobile communications [7]. Although their schemes are suitable for use in inexpensive low-power mobile communication systems, however, their schemes are not truly dynamic, where the secret information of all participants in a conference need not be modified when a participant joins or quits the conference. We have shown here that this scheme is truly dynamic in the sense that a participant joining or quitting a teleconference already in progress requires no modification of secret keys of any participants in the conference. Furthermore, the computation needed to obtain the common secret session key is relatively simple. Thus, this scheme can be used in low-power mobile communications.

One of the straightforward schemes for dynamic participation is to authenticate the conferees using NC, and also establish or reuse individual session keys for each participant. Enciphering an NC generated conference key (CK) with individual session keys for each participant is a good method. Finally, NC distributes the encrypted CK to each participant over a public channel. There are two advantages in this straightforward scheme. One is that it is easy to implement. The other is that the length of the CK is 64 b, if NC uses a symmetric cryptosystem [14] for enciphering CK. However, the straightforward scheme has the following two disadvantages. One, the NC needs to encipher CK many times depending on the number of participants. Since NC can calculate encrypted CK for each participant in advance using a large computation power, this is not a serious problem. Second, since each participant has a different encrypted CK in the straightforward scheme, NC's need to distribute the encrypted CK to each participant. Although the length of CK in the straightforward scheme is 64 b, however, NC's must distribute 64m b in the communication channel, if m participants join the secure conference. However, since each participant has the same encrypted CK in our scheme, we can broadcast the encrypted CK to each participant. Distributing the encrypted CK is required only once.

Since the main idea of the proposed scheme is to broadcast an encrypted CK to each participant, our scheme is unsuitable to use in ring configuration. However, mobile communication systems may be regarded as star-type networks [4], and it is suitable to distribute CK over a public broadcast channel. Since both the lengths Q and R are 256 b and the length of y is $\ln 256m$, the length of encrypted CK (PI) of our scheme is only $512 + \ln 256m$ b, if m participants join the secure conference. In addition, this scheme for conference key distribution for mobile communication systems enables a group of conferees to dynamically generate a common secret key over a public channel to hold a secure conference.

VII. Conclusions

We have proposed a scheme to hold secure teleconferences with digital mobile communications. The protocol enables a hardware-limited user terminal to obtain a common secret conference key in a reasonable time. Both user authentication and key distribution are included simultaneously in this scheme. We show that our scheme allows a participant to join dynamically and to quit a secure teleconference already in progress. The participants need not alter their secret information when a participant joins late or quits the conference early.

Acknowledgment

The author wishes to thank many anonymous referees for their suggestions to improve this paper.

References

[1] S. R. Hall and D. P. Maher, “Closing in on wireless privacy,” *AT&T Technol.*, vol. 8, pp. 22–25, 1993.

[2] ETSI, “Digital European cordless telecommunications common interface part 7: Security features,” Tech. Rep. Version 5.03, European Telecommunications Standards Institute, ETSI, May 1991.

[3] M. J. Beller, L. F. Chang, and Y. Yacobi, “Privacy and authentication on a portable communications system,” *IEEE J. Select. Areas Commun.*, vol. 11, pp. 821–829, Aug. 1993.

[4] M. Tatebayashi, N. Matsuzaki, and Jr. D. B. Newman, “Key distribution protocol for digital mobile communication systems,” in *Advances in Cryptology, Proc. Crypto’89*, pp. 324–334.

[5] T. Hwang, “Scheme for secure digital mobile communications based on symmetric key cryptography,” *Inform. Processing Lett.*, vol. 48, pp. 35–37, 1993.

[6] M. J. Beller, L. F. Chang, and Y. Yacobi, “Privacy and authentication on a portable communications system,” in *IEEE Globecom’91*, Phoenix, AZ, Dec. 1991, pp. 1922–1927.

[7] M. S. Hwang and W. P. Yang, “Conference key distribution protocols for digital mobile communication systems,” *IEEE J. Select. Areas Commun.*, vol. 13, pp. 416–420, Feb. 1995.

[8] C. Park, K. Kurosawa, T. Okamoto, and S. Tsujii, “On key distribution and authentication in mobile radio networks,” in *Advances in Eurocryptology, Proc. Eurocrypt’93*, 1993, pp. 131–138.

[9] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public key cryptosystems,” *Commun. ACM*, vol. 21, pp. 120–126, Feb. 1978.

[10] M. E. Smid and D. K. Branstad, “The data encryption standard: Past and future,” *Proc. IEEE*, vol. 76, pp. 550–559, May 1988.

[11] L. Gong, “A security risk of depending on synchronized clocks,” *ACM Operating Syst. Rev.*, vol. 26, no. 1, pp. 49–53, Jan. 1992.

[12] D. L. Mills, “Internet time synchronization: The network time protocol,” *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, 1991.

[13] ——, “Precision synchronization of computer network clocks,” *ACM Computer Commun. Rev.*, vol. 24, pp. 28–43, 1994.

[14] B. Schneier, *Applied Cryptography*. New York: Wiley, 1994.

[15] D. E. Knuth, *The Art of Computer Programming, Vol. 2 (Seminumberical Algorithm)*, 2nd ed. Reading, MA: Addison-Wesley, 1980.

[16] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*. Reading, MA: Addison-Wesley, 1974.

[17] G. I. Davida, D. L. Wells, and J. B. Kam, “A database encryption system with subkeys,” *ACM Trans. Database Syst.*, vol. 6, pp. 312–328, June 1981.

[18] A. Gupta and H. M. D. Toong, “An architectural comparison of 32-bit microprocessors,” *IEEE Micro*, vol. 3, pp. 9–22, 1983.

[19] ——, “Microprocessors—The first twelve years,” *Proc. IEEE*, vol. 71, pp. 1236–1256, 1983.

[20] I. Ingemarsson, D. T. Tang, and C. K. Wong, “A conference key distribution system,” *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 714–720, Sept. 1982.

[21] K. Koyama and K. Ohta, “Security of improved identity-based conference key distribution systems,” in *Lecture Notes in Computer Science*, vol. 330. Berlin, Germany: Springer, 1989, pp. 11–19.

[22] K. Koyama, “Secure conference key distribution schemes for conspiracy attack,” in *Lecture Notes in Computer Science*, vol. 658. Berlin, Germany: Springer, 1992, pp. 449–453.

[23] ——, “A secure and efficient conference key distribution system,” in *Lecture Notes in Computer Science*, vol. 950. Berlin, Germany: Springer, 1995, pp. 275–286.

[24] S. Hirose and K. Ikeda, “A conference key distribution system for the star configuration based on the discrete logarithm problem,” *Inform. Processing Lett.*, vol. 62, pp. 189–192, Sept. 1997.

Min-Shiang Hwang (S’92–M’95) received the B.S. degree in electronic engineering from National Taipei Institute of Technology, Taipei, Taiwan, R.O.C., in 1980, the M.S. degree in industrial engineering from National Tsing Hua University, Taiwan, in 1988, and the Ph.D. degree in computer and information science from National Chiao Tung University, Taiwan, in 1995. He also studied applied mathematics at National Cheng Kung University, Taiwan, from 1984 to 1986.

He passed the National Higher Examination in the electronic engineering field in 1988 and the National Telecommunication Special Examination in the information engineering field where he qualified as an Advanced Technician in 1990. From 1988 to 1991, he was the Leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications, Taiwan. He was also a Project Leader for research in computer security at TL in July 1990. He is currently an Associate Professor in the Department of Information Management, Chaoyang University of Technology, Taiwan. His current research interests include network and communication security, mobile communications, cryptography, and image compression.

Dr. Hwang is a member of ACM and the Chinese Information Security Association.