Comments on "Dynamic Participation in a Secure Conference Scheme for Mobile Communications"

Siaw-Lynn Ng

September 18, 2000

Abstract

In [1] a conference scheme is proposed for providing dynamic participation. We show that this scheme is not secure against eavesdropping and impersonation.

1 Introduction

In [1] a conference scheme is proposed for providing dynamic participation. We show that this scheme is not secure against eavesdropping and impersonation. We describe briefly the scenario and the security goals of the protocol before discussing the weaknesses.

The network centre (NC) is a trusted central authority responsible for authenticating the participants and generating and distributing session keys. The participants are U_1, the chairperson initiating the conference, and $U_2, \ldots, U_m$. Each user U_i is assumed to be in secure possession of a terminal T_i which has a reliable clock. Each U_i shares a secret $s_i = f(ID_i)$ with the network centre NC, where f is a secret one-way function known only to NC. The network centre NC has a unique identity ID_{NC} while user U_i has unique identity ID_i. The users are also assumed to have a reliable copy of NC's RSA public key (n, e). The public exponent e is assumed to be small.

The aims of the protocol are as follows. A chairperson initiates the conference. The protocol should enable three or more participants to hold a secure conference achieving the following objectives:

1. privacy of conversation content during conference;
2. privacy of participants' locations during conference;
3. prevention of fraud by ensuring portable units are authentic;
4. prevention of replay attack.

In addition, the protocol should enable dynamic joining or quitting a conference for any participants. A participant who has left should not be able to eavesdrop on the subsequent content of the conference.

2 The Protocol

The conference key distribution protocol messages proposed in [1] are as follows. The actions are explained in the following paragraphs.

$$\begin{array}{lll} (1) & T_1 \to NC: & (t_1||s_1||r_{11}||r_{12}||ID_1||\cdots||ID_m)^e \bmod n \\ (2) & NC \to T_i: & T_i \\ (3) & T_i \to NC: & (t_i||s_i||r_{i1}||r_{i2}||ID_i)^e \bmod n \\ (4) & NC \to T_i: & Q, y, R, PA \end{array}$$

In step (1), the initiator T_1 chooses random numbers r_{11}, r_{12}, each 256 bits long and computes $r_1 = r_{11} + r_{12}$ as the session key decryption key. The message sent to NC is encrypted using NC's public key and includes a timestamp t_1 and the secret s_1 shared between U_1 and NC.

In step (2) NC decrypts the message and checks that t_1 is valid, that $f(ID_1) = s_1$, and computes $r_1 = r_{11} + r_{12}$. After that NC notifies $T_2, \ldots, T_m$ of the conference.

In step (3), each T_i, $i = 2, \ldots, m$ chooses two 256-bit random numbers r_{i1}, r_{i2} and computes $r_i = r_{i1} + r_{i2}$ as the session key decryption key. As in (1), the message sent by each T_i to NC is encrypted and includes a timestamp t_i and the secret s_i shared between T_i and NC.

In step (4), after checking the validity of t_i and s_i, NC chooses a non-zero 255-bit random number CK as the session key, and a random number r_0. Then NC computes public information $PI = CK + lcm(r_0, \ldots, r_m)$ and $PA = E_{CK}(ID_{NC})$, where $E_K()$ denotes a symmetric encryption using key K. After that NC finds y and two 256-bit numbers Q, R such that $PI = Q \cdot 2^y + R$ (we will comment on this in the next section), and broadcasts Q, y, R and PA to T_i, $i = 1, \ldots, m$, and T_i obtains $CK = (Q \cdot 2^y + R) \bmod r_i$ and verifies that $E_{CK}(ID_{NC}) = PA$.

When a participant joins or leaves the conference, the other participants' secret keys (the r_is) are not modified. Only the public information PI and the secret session key CK is modified as follows:

When a participant U_{m+1} joins a conference in progress, steps (3) and (4) of the protocol are executed as follows except that $Q \cdot 2^y + R = PI = CK + r_{m+1}$. In this case CK remains unchanged for all other current participants.

When a participant U_j quits a confrence in progress, NC chooses new random numbers CK' and r'_0, and computes $PI' = CK' + lcm(r'_i, i = 0, 1, \ldots, m, i \neq j)$ where $r'_i = r_i + t'$, t' being the current timestamp. New Q', R', y' such that $PI' = Q' \cdot 2^{y'} + R'$ are computed and broadcast to T_i, $i = 1, \ldots, m, i \neq j$, together with t'. The T_is then obtain CK' by computing $(Q' \cdot 2^{y'} + R') \bmod (r_i + t') = CK'$.

3 The Attack

We show that this protocol is insecure against eavesdropping once a participant joins a conference in progress. Before that we examine the representation of PI by Q, y and R. Each r_i is a sum of two 256-bit numbers r_{i1} and r_{i2}. Therefore

r_i is a 257-bit number, and $PI = CK + lcm(r_0, \ldots, r_m)$ has at most $257(m+1)$ bits, not $256(m+1)$ bits, as claimed in [1]. Now, since Q and R are both 256-bit numbers, there will be cases when PI cannot be represented by Q, y and R as claimed in the paper. However, since the representation of PI has no bearing on the security of the protocol, we suppose that PI is broadcast as it is to the participants, and we show that the protocol is insecure nonetheless.

For ease of notation and without loss of generality, suppose that U_1, U_2, U_3 are already in conference using the session key CK when U_4 joins in the conference. When U_4 joins, NC sends $PI = CK + r_4$ to U_4, which is observed by U_1, U_2 and U_3. Now, U_1, U_2 and U_3 all know CK and hence can obtain r_4 by computing $PI - CK$. If U_3, say, should quit the conference after U_4 joined, NC would broadcast a new session key $CK' + lcm(r'_0, r_1 + t', r_2 + t', r_4 + t')$ and t'. However, U_3 would then be able to compute CK' because he knows t' and r_4 and hence would be able to eavesdrop on the conference that he has just left. Similarly, a new participant U_5 may also follow the same line of attack by waiting for a participant to join before joining the conference himself and leaving immediately while continuing to be able to eavesdrop. This attack works because the new participant's secret is only masked by an entity CK which is known to all other current participants. One way of stopping this attack is to send $CK + r_i \cdot k_i$ for some random number k_i, so that r_i is never known by another participant.

4 Conclusion

The protocol of [1] is designed to be dynamic. Therefore it is important that the participants currently in conference should be known to all the participants. However the only entity with this knowledge is NC. The participants are authenticated by NC, and two participants U_i, U_j accept each other as authorised participants on the grounds that only authenticated participants know the session key CK. However, because of the attack described above, this premise does not hold and hence this conference scheme is insecure against impersonation as well as eavesdropping.

References

[1] M.-S. Hwang, "Dynamic participation in a secure conference scheme for mobile communications", *IEEE Transactions on Vehicular Technology*, Vol. 48, No. 5, September 1999, pp. 1469–1474.