Fixing a problem in the Helsinki protocol

Chris J. Mitchell and Chan Yeob Yeun
Information Security Group,
Royal Holloway, University of London,
Egham, Surrey TW20 0EX, UK
{c.mitchell,c.yeun}@rhbnc.ac.uk

24th June 1998

Abstract

We consider a recently described attack on a key establishment protocol contained in a draft international standard. Based on an observation as to why the attack is possible, we propose a simple modification to the protocol which avoids the attack.

1 Introduction

Horng and Hsu, [2], have shown how a attack can be launched on a key establishment protocol they call the *Helsinki Protocol*. This protocol is designed to establish a shared secret key between two entities A and B, and is specified as Key Transport Mechanism 6 in Clause 7.6 of ISO/IEC DIS 11770–3, [3]. It is claimed in [3] that this protocol provides mutual entity authentication and mutual key confirmation, i.e. both A and B have confirmation that the other party has a copy of the shared key.

However, if the attack is successfully carried out by a malicious third party C, then B believes it has authenticated and established a shared secret key with A, whereas A believes it has authenticated and established a (different) shared secret key with C. This means that the claim of mutual key confirmation is incorrect, and the claim of mutual authentication is at best highly suspect.

Before proceeding observe that the Helsinki protocol is actually a derivative of a protocol originally described by Needham and Schroeder in 1978, [7]. It also embodies features from the COMSET protocol, which was devised as part of the RIPE project, [1]. For further information see Sections 12.5.1 and 12.10 of [6].

Also note that the Horng-Hsu attack is closely related to the Lowe attack on the Needham-Schroeder protocol, [4, 5]. Moreover, the modification we propose below to the Helsinki protocol corresponds directly to the modifications Lowe proposes to the Needham-Schroeder protocol. In [5] Lowe proves that his modified Needham-Schroeder protocol is secure (within a specified formal model), giving added confidence that the modified version of the Helsinki protocol is sound.

2 The Protocol

The protocol in question involves the exchange of three messages between A and B. The protocol requires A and B to have an agreed public key encryption scheme, and to have their own encryption/decryption key pairs for this scheme. We also assume that A and B have (reliably) exchanged their public keys. The protocol messages are as follows.

M_1: $A \rightarrow B$: $E_B(I_A||K_A||r_A)$

M_2: $B \rightarrow A$: $E_A(K_B||r_A||r_B)$

M_3: $A \rightarrow B$: r_B

where $E_X(Y)$ denotes the public key encryption of data Y using the private encryption key of X, $X||Y$ denotes the concatenation of data items X and Y, I_X is an identifier for entity X, r_A and r_B are random 'nonces' (i.e. one-time random challenges), and K_A and K_B are key components, generated by A and B respectively.

At the end of the protocol K_A and K_B are combined using a one-way function to establish a shared secret key. Of course, to give a complete specification of the protocol we need to indicate what checks are performed by A and B during execution of the protocol, but for the sake of brevity we omit them here.

3 The Horng-Hsu attack and an observation

The attack in [2] operates as follows. C commences the attack by causing A to inaugurate a run of the protocol with C. A then sends the following message:

M_1: $A \rightarrow C$: $E_C(I_A||K_A||r_A)$

C decrypts the message to obtain r_A, and uses it to create a forged message M_1', containing a new key component K_A', which C sends to B. When sending this message, C pretends that it is from A.

M_1': $C \rightarrow B$: $E_B(I_A||K_A'||r_A)$

B responds to C (thinking it is responding to A) with the following message:

M_2: $B \rightarrow C$: $E_A(K_B||r_A||r_B)$

C intercepts this message and forwards it (unchanged) to A. A responds to C with the following message:

M_3: $A \rightarrow C$: r_B

C then forwards this message to B.

After these exchanges:

- A believes it has established a shared secret key with C, based on the key components K_A and K_B (although C does not know K_B), and
- B believes it has established a shared secret key with A, based on the key components K'_A and K_B (although A does not know K'_A).

Note that this is an example of an 'Insider attack'. This holds since, in order to launch the attack, C must persuade A to inaugurate a run of the protocol, and hence C must be an entity with whom A is prepared to establish a shared secret key.

Note also that this attack is possible since, whereas B actually generates M_2, A will believe it comes from C. This is possible because message M_2 contains no indication of its source (unlike message M_1). Hence, although C cannot discover the precise contents of message M_2, C can forward it to A and have it accepted as originating from C, although it was actually generated by B.

4 A revised version of the protocol

Based on the observation we have just made about why the attack is possible, we propose that the protocol should be modified in the following minimal way. The second message M_2 should be replaced by a modified message, which we call N_2:

N_2: $B \rightarrow A$: $E_A(I_B||K_B||r_A||r_B)$

That is, the only change is to insert an identifier for B in the second protocol message. The other two protocol messages remain unchanged.

5 Conclusions

We have describes a simple modification to the Helsinki protocol which prevents the Horng-Hsu attack, but yet which does not add significantly to the communications or computational overhead for the protocol. Note that both the original and amended protocols depend very much on an implicit property of the public key encryption scheme. Specifically, the protocols require the encryption scheme to provide a measure of integrity protection for encrypted strings.

Added note

After completion of this paper and circulation of a preprint at the April 1998 meeting, ISO/IEC JTC1/SC27/WG2 agreed that, when it is published, ISO/IEC 11770-3 will contain the modified version of the Key Transport Mechanism 6 described in this paper.

References

[1] A. Bosselaers and B. Preneel, editors. *Integrity Primitives for Secure Information Systems: Final Report of RACE Integrity Primitives Evaluation RIPE-RACE 1040.* Number 1007 in Lecture Notes in Computer Science. Springer-Verlag, New York, 1995.

[2] G. Horng and C.-K. Hsu. Weakness in the Helsinki protocol. *Electronics Letters*, **34**:354–355, 1998.

[3] International Organization for Standardization, Genève, Switzerland. *ISO/IEC 2nd DIS 11770-3, Information technology—Security techniques—Key management; Part 3: Mechanisms using asymmetric techniques*, July 1997.

[4] G. Lowe. An attack on the Needham-Schroeder public-key authentication protocol. *Information Processing Letters*, **56**:131–133, 1995.

[5] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In Margaria and Steffen, editors, *Tools and algorithms for the construction and analysis of systems*, number 1055 in Lecture Notes in Computer Science, pages 147–166. Springer-Verlag, Berlin, 1996.

[6] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, Boca Raton, 1997.

[7] R.M. Needham and M.D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, **21**:993–999, 1978.