On Establishing and Fixing a Parallel Session Attack in a Security Protocol

Reiner Dojen[1], Anca Jurcut[1], Tom Coffey[1], and Cornelia Gyorodi[2]

[1] Department of Electronic & Computer Engineering, University of Limerick, Ireland. E-Mail: reiner.dojen@ul.ie, anca.jurcut@ul.ie, tom.coffey@ul.ie

[2] Department of Computer Science, University of Oradea, Romania. E-Mail: cgyorodi@uoradea.ro

Summary. Nowadays mobile and fixed networks are trusted with highly sensitive information, which must be protected by security protocols. However, security protocols are vulnerable to a host of subtle attacks, such as replay, parallel session and type-flaw attacks. Designing protocols to be impervious to these attacks has been proven to be extremely challenging and error prone.

This paper discusses various attacks against security protocols. As an example, the security of the Wide-Mouthed Frog key distribution protocol when subjected to known attacks is discussed. Significantly, a hitherto unknown attack on Lowe's modified version of the Wide-Mouthed Frog protocol is presented. Finally, a correction for the protocol to prevent this attack is proposed and discussed.

Key words: Security protocols, protocol flaws, parallel session attack

Acknowledgement. This work was funded by the Irish Research Council for Science, Engineering and Technology (IRCSET Embark Initiative) and Science Foundation Ireland - Research Frontiers Programme (SFI RFP07 CMSF 631).

1 Introduction

With the ever increasing use of distributed applications for meeting customer demands, there is a commensurate increase in the reliance of electronic communication over networks. Since many of these applications require the exchange of highly sensitive information over computer networks, the need for security protocols to ensure the protection and integrity of data is critical. Basic cryptographic protocols allow protocol principals to authenticate each other, to establish fresh session keys for confidential communication and to ensure the authenticity of data and services. Building on such basic cryptographic protocols, more advanced services like non-repudiation, fairness, electronic payment and electronic contract signing are achieved.

In this paper we introduce attacks against security protocols. As an example, the Wide-Mouthed Frog key distribution protocol is discussed and known attacks on the protocol are detailed. Additionally, a hitherto unknown attack on Lowe's modified version of the Wide-Mouthed Frog protocol is presented. A correction for the protocol to prevent the attack is proposed and discussed.

2 Attacks Against Security Protocols

A security protocol should enforce the data exchange between honest principals, while the dishonest ones should be denied any benefit of it. However, security protocols can contain weaknesses that make them vulnerable to a range of attacks such as replay, parallel session and type-flaw attacks. Many security protocols have been found to contain weaknesses, which have subsequently been exploited, for example [1][2][3][4][5][6][7][8][9]. This highlights the difficulty of designing effective security protocols.

A replay attack is one of the most common attacks on authentication and key-establishment protocols. If the messages exchanged in an authentication protocol do not carry appropriate freshness identifiers, then an adversary can get himself authenticated by replaying messages copied from a legitimate authentication session. Examples of replay attacks include [1][2][5][6].

A parallel session attack requires the parallel execution of multiple protocol runs, where the intruder uses messages from one session to synthesise messages in the other session. Examples of parallel session attacks on security protocols include [1][3][4][7]. There are several forms of parallel session attacks. In an oracle attack the intruder starts a new run of the protocol and uses one of the principals as an oracle for appropriate answers to challenges in first protocol run. A man-in-the-middle attack occurs when two principals believe they are mutually authenticated, when in fact the intruder masquerades as one principal in one session and as the other principal in another. A multiplicity attack is a parallel session attack where the principals disagree on the number of runs they have successfully established with each other.

A type flaw attack involves the replacement of a message component with another message of a different type by the intruder. Examples of type flaw attack on security protocols include [4][8][9].

3 The Wide-Mouthed Frog Protocol

To illustrate the difficulty of designing effective security protocols, the Wide-Mouthed Frog protocol, published by Burrows, Abadi and Needham [2], is discussed. It was proposed for the distribution of a fresh key between peer entities. It assumes the use of symmetric key cryptography, a trusted server and synchronized clocks. Anderson and Needham claimed a replay attack on the protocol [10] and another attack was discovered by Lowe, who modified the protocol to prevent these attacks[11]. Further, the authors of this paper present a hitherto unknown attack on the protocol. The presence of flaws in the protocol in spite of the repeated revisions demonstrates the difficulty of designing effective security protocols.

Fig. 1 depicts the steps of the Wide-Mouthed Frog protocol: In step 1 A sends timestamp Ta, the identity of B and the new session key Kab, encrypted with Kas to server S. The server forwards the session key, along with A's identity and timestamp Ts to B using Kbs. Table 1 explains the used notation.

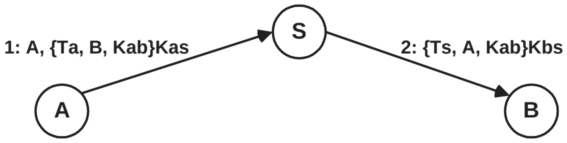

Fig. 1. The Wide-Mouthed Frog Protocol

{m}k	message m encrypted with key k
A,B	legitimate principals
S	trusted server
Kas/Kbs	symmetric key shared between A & S and B & S
I(A)/I(B)	intruder masquerading as principal A/B
Ta/b/s	timestamp created by A/B/S
Na/b	once off random number (nonce) generated by principal A/B

Table 1. Notation for Protocol Descriptions

To-date, two attacks on the Wide-Mouthed Frog protocol have been published. Anderson and Needham claim a replay attack [10] and Lowe a multiplicity attack [11].

3.1 A Replay Attack on the Wide-Mouthed Frog Protocol

The replay attack on the Wide-Mouthed Frog protocol enables an attacker to keep the session key Kab alive [10]. It assumes that the server does not keep a list of all recent keys.

The attack - depicted in Fig. 2 - works as follows: After intercepting the message exchange of the protocol run i, the intruder starts a new run of the protocol (ii) masquerading as B and sends to the server as message ii.1 a copy of message i.2. Intercepting message ii.2, the intruder can start run iii of the protocol and masquerade as A and so on. As a result the session key Kab is kept alive by the intruder.

3.2 A Multiplicity Attack on the Wide-Mouthed Frog Protocol

Although Burrows, Abadi, and Needham claimed that both time stamps and nonces ensure freshness of message exchanges in the Wide-Mouthed Frog pro-

```
i.1. A -> S : A, {Ta, B, Kab}Kas
i.2. S -> B : {Ts, A, Kab}Kbs
   ii.1. I(B) -> S : B, {Ts, A, Kab}Kbs
   ii.2. S -> A : {Ts', B, Kab}Kas
        iii.1. I(A) -> S : A, {Ts', B, Kab}Kas
        iii.2. S -> B : {Ts'', A, Kab}Kbs
        ....
```

Fig. 2. Replay attack on the Wide-Mouthed Frog Protocol

tocol [2], Lowe demonstrated that time stamps only ensure recentness of a message [11]. The protocol is thus susceptible to a multiplicity attack.

In this attack an intruder records the second message and replays it to B. Alternatively, the intruder could replay the first message to the server. As a result, B is lead into thinking that A is trying to establish a second session, whereas A has established only one session. The attack involves two interleaved runs of the protocol, as shown in Fig. 3. For this attack to succeed it is assumed that the principals do not keep lists of recent keys.

```
i.1. A -> S : A, {Ta, B, Kab}Kas
i.2. S -> B : {Ts, A, Kab}Kbs
   ii.2. I(S) -> B : {Ts, A, Kab}Kbs
```

Fig. 3. Multiplicity Attack on the Wide-Mouthed Frog Protocol

3.3 Lowe's Modified Wide-Mouthed Frog Protocol

In Lowe's modified version of the Wide-Mouthed Frog protocol [11] a nonce handshake between A and B (cf. Fig. 4) has been added to prevent the presented attacks. To-date no weaknesses of Lowe's modified versions of the Wide-Mouthed Frog protocol have been published.

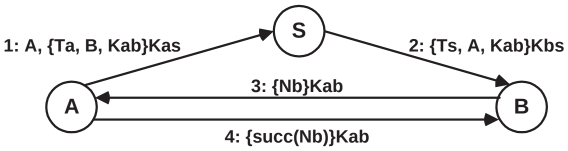

Fig. 4. Lowe's Modified Wide-Mouthed Frog Protocol

4 A New Attack on Lowe's Modified Protocol

We now present a hitherto unknown parallel session attack on Lowe's modified Wide-Mouthed Frog protocol, which uses A as an oracle. Consequently, the intruder impersonates principal B to A. Fig. 5 details the attack, which assumes that A initiates the protocol. The intruder intercepts message i.2 and starts a new protocol run as B, using message i.2 as part of ii.1. The honest principal A decrypts message ii.2, extracts key Kab and responds by sending message ii.3 which contains a freshly generated nonce Na encrypted with this session key. The intruder then replays ii.3 as its next message in run i (i.3.). In response, A returns the encrypted successor of Na (i.4.), which the intruder replays as final message in run ii (ii.4).

```
i.1. A -> S: A,{Ta, B, Kab}Kas
i.2. S -> I(B): {Ts, A, Kab}Kbs
    ii.1.  I(B) -> S: B, {Ts, A, Kab}Kbs
    ii.2. S -> A: {Ts’, B, Kab}Kas
    ii.3. A -> I(B) : {Na}Kab
i.3. I(B) -> A: {Na}Kab
i.4. A -> I(B): {succ(Na)}Kab
    ii.4. I(B) -> A: {succ(Na)}Kab
```

Fig. 5. New Attack on Lowe’s Modified Wide-Mouthed Frog Protocol

Consequently, A believes that a session has been established with B in run i and also believes that B has established a session in run ii, even though B is in fact absent. Therefore, the protocol does not ensure authentication.

The protocol suffers from two weaknesses: One is the symmetry of the first two encrypted messages of the protocol, which allows the intruder to replay the second message of the first protocol run (i.2) as the initial message of the second run (ii.1). The other and major weakness is that it is not possible to establish the source of the second last message.

4.1 Fixing the Flaw

In order to fix the protocol, the source of message 3 needs to be established. This can be achieved by including the sender’s identity in the third message of the protocol as presented in Fig. 6.

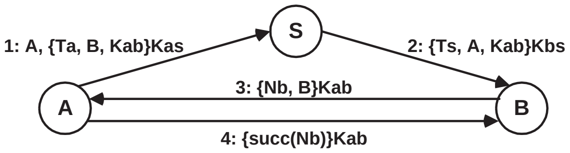

Fig. 6. Proposed Correction to Wide-Mouthed Frog Protocol

As message 3 now includes the encrypted identity of B (originator of the message), an intruder cannot successfully replay the message in a parallel session with switched roles of initiator and responder: Message ii.3 in Fig. 5 becomes {`Na, A`}`Kab`, whereas message i.3 requires {`Na, B`}`Kab`. Thus, the presented attack is prevented. Further, formal verification of the proposed protocol using the OFMC tool [12] provides confidence in its correctness.

5 Conclusion

A security protocol should enforce the data exchange between honest principals, while the dishonest ones should be denied any benefit of it. However, security protocols can contain weaknesses that make them vulnerable to a range of attacks, such as replay, parallel session and type-flaw attacks. Designing protocols to be resistant against these attacks has been proven very challenging as is highlighted by the large number of security protocols that have been found to contain exploitable weaknesses.

This paper analysed the Wide-Mouthed Frog key distribution protocol and known attacks on the protocol were presented. Additionally, a hitherto unknown attack on Lowe's modified version of the Wide-Mouthed-Frog protocol that allows an intruder to impersonate an legitimate principal was detailed. A correction for the protocol to prevent the attack was proposed and discussed.

References

1. C. Boyd, A. Mathuria, Protocols for authentication and key establishment, Berlin, London, Springer-Verlag, 2003.
2. M. Burrows, M. Abadi, R. Needham. A logic of authentication. ACM Transactions on Computer Systems TOCS, 8(1):18-36, 1990.
3. G. Lowe. An attack on the Needham-Schroeder public key authentication protocol. Information Processing Letters, 56(3):131-136, November 1995.
4. G. Lowe. Some new attacks upon security protocols. IEEE Computer Society Press, Proceedings of Computer Security Foundations Workshop VIII, 1996.
5. D. Denning, G. Sacco. Timestamps in key distributed protocols. Communication of the ACM, 24(8):533-535, 1981.
6. T. Aura. Strategies against replay attacks.Proceedings of the 10th IEEE Computer Society Foundations Workshop, pp.59-68, Rockport, MA, June 1997.
7. J. Nam, S. Kim, S. Park, D. Won: Security analysis of a nonce-based user authentication scheme using smart cards. IEICE Transactions Fundamentals, 90(1):299-302, 2007.
8. T. Hwang, N.Y. Lee, C.M. Li, M.Y. Ko, Y.H. Chen. Two attacks on Neumann-Stubblebine authentication protocols. Information Processing Letters, 53:103-107, 1995.
9. J. Heather, G. Lowe, S. Schneider. How to prevent type flaw attacks on security protocols. pp.255-268. IEEE Computer Society, 2000.
10. R. Anderson, R. Needham. Programming Satan's Computer. Computer Science Today - Recent Trends and Developments, Springer LNCS 1000:426-440. 1995.
11. G. Lowe. A family of attacks upon authentication protocols. Technical Report 1997/5, Dept. Mathematics & Computer Science, University of Leicester, 1997.
12. D. Basin, S. Mdersheim, L. Vigan. OFMC: A symbolic model checker for security protocols. Int. Journal of Information Security 4(3):181-208, 2005.