Wide Mouthed Frog

Author(s): Michael Burrows 1989
Last modified November 20, 2002

Summary: Distribution of a fresh shared key. Symmetric key cryptography with server and timestamps.

Protocol specification (in common syntax)

```
A, S :           principal
Kas, Kbs, Kab :  symkey
Ta, Ts :         timestamp

1.   A  ->  S   :   A, {Ta, B, Kab}Kas
2.   S  ->  B   :   {Ts, A, Kab}Kbs
```

Description of the protocol rules

Some explanations quoted from [BAN89]:

> "It is assumed that the encryption is done in such a way that we know the whole message was sent at once. If two separate encrypted sections are included in one message, we treat them as though they arrived in separate messages. A message cannot be understood by a principal who does not know the key (or, in the case of public-key cryptography, by a principal who does not know the inverse of the key); the key cannot be deduced from the encrypted message. Each encrypted message contains sufficient redundancy to allow a principal who decrypts it to verify that he has used the right key. In addition, messages contain sufficient information for a principal to detect (and ignore) his own messages."

> "`A` sends a session key to `S`, including a timestamp `Ta`. `S` checks that the first message is timely, and if it is, it forwards the message to `B`, together with its own timestamp `Ts`. `B` then checks that the timestamp from `S` is later than any other it has received from `S`."

Requirements

The protocol must guaranty the secrecy of the new shared key `Kab`: in every session, the value of `Kab` must be known only by the participants playing the roles of `A` and `B` and `S`.

The protocol must guaranty the authenticity of `Kab`: in every session, on reception of message `2`, `B` must be ensured that the key `Kab` in the message has been created by `S` in the same session on behalf of `A`.

References

[BAN89]

Claimed proofs

[BAN89]

Claimed attacks

1. [AN95]. By replaying the second message within an appropriate time window, the intruder `I` can make the server `S` update the timestamp of an non-fresh key `Kab`. This way, he can extend the life time of a (possibly compromised) key `Kab` as wanted, whereas `A` and `B` think that it has expired and has been destroyed.

```
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  :  {T's, B, Kab}Kas
iii.1.    I(A) -> S  :  A, {T's, B, Kab}Kas
iii.2.    S    -> B  :  {T''s, A, Kab}Kbs
....
```

2. [Low97]. In this attack, `B` thinks that `A` has established two sessions with him, when `A` thinks he has established only one session.

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

[Low97] proposes a correction of the protocol which is described in Lowe modified Wide Mouthed Frog.

Comment sent by Martin Abadi (*November 18, 2002*)

The [AN95] and [Low97] "attacks" fail, because of the protocol features described in the quotations above. The "attacks" may work only against (deliberately or unintentionally) weakened variants of the protocol.

See also

Lowe modified Wide Mouthed Frog

Citations

[AN95] R. Anderson and R. Needham. Programming satan's computer, 1995.

[BAN89] Michael Burrows, Martin Abadi, and Roger Needham. A logic of authentication. Technical Report 39, Digital Systems Research Center, february 1989.

[Low97] Gavin Lowe. A family of attacks upon authentication protocols. Technical Report 1997/5, Department of Mathematics and Computer Science, University of Leicester, 1997.