Bull's Authentication Protocol

Author(s): J. Bull 1997

Summary: This protocol, described in [BO97], aims at establishing fresh session keys between a fixed number of participants (for instance 3) and a server: one key for each pair of agents adjacent in the chain.

Protocol specification (in common syntax)

```
A, B, C, S :       principal
Kab, Kbc :         fresh symkey
Na, Nb, Nc :       fresh number
Kas, Kbs, Kcs :    symkey
h :                message, symkey -> message

A computes Xa = h((A,B,Na),Kas), (A,B,Na)

1.   A  ->  B   :   Xa

B computes Xb = h((B,C,Nb,Xa),Kbs), (B,C,Nb,Xa)

2.   B  ->  C   :   Xb

C computes Xc = h((C,S,Nc,Xb),Kcs), (C,S,Nc,Xb)

3.   C  ->  S   :   Xc

4.   S  ->  C   :   A, B, Kab xor h(Na,Kas), {A,B,Na}Kab,
                    B, A, Kab xor h(Nb,Kbs), {B,A,Nb}Kab,
                    B, C, Kbc xor h(Nb,Kbs), {B,C,Nb}Kbc,
                    C, B, Kbc xor h(Nc,Kcs), {C,B,Nc}Kbc
5.   C  ->  B   :   A, B, Kab xor h(Na,Kas), {A,B,Na}Kab,
                    B, A, Kab xor h(Nb,Kbs), {B,A,Nb}Kab,
                    B, C, Kbc xor h(Nb,Kbs), {B,C,Nb}Kbc
6.   B  ->  A   :   A, B, Kab xor h(Na,Kas), {A,B,Na}Kab
```

Description of the protocol rules

The protocol is initiated by `A` and then goes through `B` and `C` before reaching `S`. At the end, new session keys `Kab` and `Kbc` are established. The properties of exclusive or are:

$$\begin{aligned} \texttt{x xor (y xor z)} &= \texttt{(x xor y) xor z} \quad \text{(E1)} \\ \texttt{x xor y} &= \texttt{y xor x} \quad \text{(E2)} \end{aligned}$$

$$\begin{array}{rcll} \texttt{x xor 0} & = & \texttt{x} & \text{(E3)} \\ \texttt{x xor x} & = & \texttt{0} & \text{(E4)} \end{array}$$

Requirements

The protocol must guaranty the secrecy of Kxy. Each key Kxy should be known to exactly x and y (and also S), even if some nodes other than x and y are malicious.

References

[BO97]

Claimed attacks

This protocol is subject to an attack [RS98] that can be mounted by a dishonest participant. For example, assume that C is a malicious agent. He can intercept Kab xor h(Nb,Kbs) and Kbc xor h(Nb,Kbs) sent by S at step 4, and since C knows the session key Kbc, he can compute Kbc xor Kab xor h(Nb,Kbs) xor Kbc xor h(Nb,Kbs). Since this term is actually equal to Kab, the agent C learns a session key that should be shared only by A and B.

Citations

[BO97] J. Bull and D. J. Otway. The authentication protocol. Technical Report DRA/CIS3/PROJ/CORBA/SC/1/CSM/436-04/03, Defence Research Agency, 1997.

[RS98] P. Y. A. Ryan and S. A. Schneider. An attack on a recursive authentication protocol: A cautionary tale. *Information Processing Letters*, 65(1):7–10, 1998.