Kerberos V5

Author(s): B. Clifford Neuman and Theodore Ts'o 1994
Last modified November 7, 2002

Summary: Distribution of a symmetric key (in a *ticket*), for communication between a client and a server, with authentication.

Remark

This protocol is based on based on the Needham Schroeder Symmetric Key protocol and uses timestamps and nonces to correct the flaw of Denning Sacco.

Protocol specification (in common syntax)

```
A, G, C, S, U :           principal
N1, N2 :                  nonce
L1, L2 :                  nonce
T1start, T1expire :       timestamp
T2start, T2expire :       timestamp
Kcg, Kcs, Kag, Ku, Kgs :  key

1.    C  ->  A  :   U, G, L1, N1
2.    A  ->  C  :   U, {U, C, G, Kcg, T1start, T1expire}Kag,
                    {G, Kcg, T1start, T1expire}Ku
3.    C  ->  G  :   S, L2, N2, {U, C, G, Kcg, T1start, T1expire}Kag,
                    {C, T1}Kcg
4.    G  ->  C  :   U, {U, C, S, Kcs, T2start, T2expire}Kgs,
                    {S, Kcs, T2start, T2expire, N2}Kcg
5.    C  ->  S  :   {U, C, S, Kcs, T2start, T2expire}Kgs,
                    {C, T2}Kcs
6.    S  ->  C  :   {T2}Kcs
```

Description of the protocol rules

`C` is a client,

`S` is a a server (`C` wants to communicate with `S`),

`U` is a user on behalf of which `A` and `S` communicate,

`G` is a ticket granting server,

`A` is a key distribution center (trusted server).

The keys `Kag` and `Kgs` are long term symmetric key whose values are supposed to be known initially only by, `A` and `G`, respectively `G` and `S`.

`L1` and `L2` are lifetimes, `N1` and `N2` are nonces. `T1start`, `T1expire`, `T2start`, `T2expire` are time stamps which define the interval of validity of the ticket in which they are contained.

`U` is a user on behalf of whom the client `C` communicates. In particular, `C` initially knows the value of the key `Ku`.

The key `Kcg` is freshly generated by `A` for communication between `C` and `G`, and in transmitted to `C` in message `2`, encrypted by `Ku`, and indirectly to `G`, in the *ticket* {`U, C, G, Kcg, T1start, T1expire`}`Kag` which `C` transmits blindly to `G` in message `3`.

The authentificator {`C, T1`}`Kcg` is used by `G` to check timeliness of the ticket.

The key `Kcs` is freshly generated by `G` for communication between `C` and `G`, and in transmitted to `C` in message `4`, encrypted by `Kcg`, and indirectly to `S`, in the *ticket* {`U, C, S, Kcs, T2start, T2expire`}`Kgs` which `C` transmits blindly to `S` in message `5`.

Requirements

The protocol must guaranty the secrecy of `Kcs`: in every session, the value of `K` must be known only by the participants playing the roles of `A`, `B` and `S` in that session.

`A` and `C` must agree on the values of `T1start` and `T1expire`.

`G` and `C` must agree on the values of `T2start` and `T2expire` and `T1`.

`C` and `S` must agree on the value of `T2`.

References

[NT94]

Claimed proofs

- [NT94]
- [BAN89]
- [SMB90] modelization with Abstract State Machines (stepwise refinements), and (manual) proof of correctness.

See also

Needham Schroeder Symmetric Key

Citations

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

[NT94] B. Clifford Neuman and Theodore Ts'o. Kerberos : An authentication service for computer networks. Technical Report ISI/RS-94-399, USC/ISI, 1994.

[SMB90] Michael Merritt Steven M. Bellovin. Limitations of the kerberos authentication system. *Computer Communication Review*, 20(5):119–132, october 1990.