# KSL

**Author(s):** Axel Kehne and Jürgen Schönwälder and Horst Langendörfer
1992
*Last modified December 2, 2002*

**Summary:** Nonce based improvement of Kerberos V5 protocol with generalized timestamps. Distribution of a session key and a ticket and repeated mutual authentication. Symmetric key cryptography with server.

## Protocol specification (in common syntax)

```
A, B, S :               principal
Na, Nb, Nc, Ma, Mb :    number
Kas, Kbs, Kab, Kbb :    key
Tb :                    generalizedTimestamp

1.   A  ->  B  :   Na, A
2.   B  ->  S  :   Na, A, Nb, B
3.   S  ->  B  :   {Nb, A, Kab}Kbs, {Na, B, Kab}Kas
4.   B  ->  A  :   {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, Nc, {Na}Kab
5.   A  ->  B  :   {Nc}Kab

6.   A  ->  B  :   Ma, {Tb, A, Kab}Kbb
7.   B  ->  A  :   Mb, {Ma}Kab
8.   A  ->  B  :   {Mb}Kab
```

## Description of the protocol rules

The messages `1-5` are the part concerning the generation and exchange of the session key `Kab`. The messages `6-8` are for mutual authentification. This second part of the protocol is also called *repeated authentication* because it can be repeated alone several times, until the ticket `{Tb, A, Kab}Kbb` expires.

**Key exchange.** The keys `Kas` and `Kbs` are long term symmetric key whose values are supposed to be known initially only by `A` and `S`, respectively `B` and `S`.

The session key `Kab` is freshly generated by `S` and in sent in message `3` directly to `B`, and indirectly to `A`, in the cipher `{Na, B, Kab}Kas`, transmitted blindly to `A` by `B` in message `4`.

Kbb is a secret key only known to B, used to encrypt the ticket {Tb, A, Kab}Kbb in message 4. This ticket will be used in the repeated authentication.

**Repeated authentication.** In the ticket {Tb, A, Kab}Kbb, Tb is a generalized timestamp, made of a timestamp from the local clock of B, a lifetime limiting the validity of the ticket (relatively to the local clock of B) and a clock identifier, i.e. a nonce which is updated each time B's local clock is corrected.

When he receives a ticket in message 6, B compares the time identifier in Tb to the current identifier of his local clock and if they match, verifies the validity of the ticket, i.e. he checks that the time of his local clock is within the time window defines by the timestamp and the lifetime of Tb. If one of these tests fails, then B rejects the ticket. Otherwise, he starts an exchange of nonces (messages 7 and 8) the purpose of which is to convince mutually A and B that they both possess the session key Kab.

## Requirements

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

The protocol must also ensures mutual authentication of A and B.

## References

[KSL92]

## Claimed proofs

The authors of the protocol propose in [KSL92] an analysis in the framework of the BAN logic [BAN89].

## Claimed attacks

**1.** [Low96]: "The repeated authentification part can be used as an encrypting oracle". If the intruder I wants to encrypt some data M with the session key Kab, he can run:

```
6.   I(A)  ->   B    :   M, {Tb, A, Kab}Kbb
7.    B    ->  I(A)  :   Mb, {M}Kab
```

The ticket {Tb, A, Kab}Kbb can have been learned by I from the message 4 of a previous key distribution. After running the two above message, I can send:

`I(A)  ->  B  :  {M}Kab` and B will accept this message as having been sent by A.

**2.** The attacks presented in [HLL+95] on the repeated authentication part of the neumannStubblebine protocol also works here.

This attack concerns the repeated authentication part, assuming Kab has been recorded in a previous legitimate run of the protocol.

```
i.6.    I(A)  ->   B    :   Mi, { Tb, A, Kab}Kbb
i.7.     B    ->  I(A)  :   Mb, {Mi}Kab
ii.6.   I(A)  ->   B    :   Mb, {Tb, A, Kab}Kbb
ii.7.    B    ->  I(A)  :   Mb', {Mb}Kab
i.8.    I(A)  ->   B    :   {Mb}Kab
```

**3.** [Low96]. In this scenario, two tickets generated by two different agents contains the same session key Kab, which, according to [Low96], was supposed not to happen in the protocol of [KSL92].

```
i.1.    I(A)  ->   B    :   Ni, A
i.2.     B    ->  I(S)  :   Ni, A, Nb, B
ii.1.   I(B)  ->   A    :   Nb, B
ii.2.    A    ->   S    :   Nb, B, Na, A
ii.3.    S    ->   A    :   {Na, B, Kab}Kas, {Nb, A, Kab}Kbs
ii.4.    A    ->  I(B)  :   {Nb, A, Kab}Kbs, {Ta, B, Kab}Kaa, Nc, {Nb}Kab
i.3.    I(S)  ->   B    :   {Nb, A, Kab}Kbs, {Na, B, Kab}Kas
i.4.     B    ->  I(A)  :   {Na, B, Kab}Kas, {Tb, A, Kab}Kbb, Nc', {Ni}Kab
```

The intruder I can then use the two tickets to complete step 5 of both runs i and ii. In this scenarion, the repeated authentication procedure is used as an encrypting oracle.

```
i.6.    I(A)  ->   B    :   Nc, {Tb, A, Kab}Kbb
i.7.     B    ->  I(A)  :   Mb, {Nc}Kab
ii.5.   I(B)  ->   A    :   {Nc}Kab
ii.6.   I(B)  ->   A    :   Nc', {Ta, B, Kab}Kaa
ii.7.    A    ->  I(B)  :   Ma, {Nc'}Kab
i.5.    I(A)  ->   B    :   {Nc'}Kab
```

**4.** [Low96]. The ticket obtained in the first part of the above scenario also permits I to impersonate A in the repeated authentification part of the protocol.

```
i.6.     I(A)  ->    B    :    Mi, {Tb, A, Kab}Kbb
i.7.       B   ->  I(A)   :    Mb, {Mi}Kab
ii.6.    I(B)  ->    A    :    Mb, {Ta, B, Kab}Kaa
ii.7.      A   ->  I(B)   :    Ma, {Mb}Kab
i.8.     I(A)  ->    B    :    {Mb}Kab
```

### See also

Kerberos V5, Neumann Stubblebine, Lowe modified KSL.

## Citations


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

[HLL+95] Tzonelih Hwang, Narn-Yoh Lee, Chuang-Ming Li, Ming-Yung Ko, and Yung-Hsiang Chen. Two attacks on neumann-stubblebine authentication protocols. *Information Processing Letters*, 53:103 – 107, 1995.

[KSL92] Axel Kehne, Jürgen Schönwälder, and Horst Langendörfer. Multiple authentications with a nonce-based protocol using generalized timestamps. In *Proc. ICCC '92*, Genua, 1992.

[Low96] Gavin Lowe. Some new attacks upon security protocols. In IEEE Computer Society Press, editor, *In Proceedings of the Computer Security Foundations Workshop VIII*, 1996.