Kerberos: a Session Key Distribution Scheme

For Session Key Distribution Schemes (SKDS), we will assume that there is a network of users with a trusted authority (TA), there is an agreed-upon symmetric key encryption algorithm $e$, and that each user shares a secret key with the TA that can be used on $e$. For Alice and Bob, we shall denote their shared secret key with the TA as $K_{Alice}$ and $K_{Bob}$ respectively.

What is an SKDS? Suppose Alice wants to talk to Bob. An SKDS is a protocol that describes how Alice and Bob can obtain a TA-issued session key which they then can use to communicate with each other.

The Needham-Schroeder Scheme

One of the first SKDS was proposed by Needham and Schroeder in 1978. It works as follows:

1. Alice chooses a random number $r_A$. She sends $ID(Alice)$, $ID(Bob)$ and $r_A$ to the TA.

2. The TA chooses a random session key $K$. It then computes a two values that contain $K$. The “ticket” for Bob is

   $$t_{Bob} = e_{K_{Bob}}(K||ID(Alice))$$

   while the value for Alice is

   $$y_1 = e_{K_{Alice}}(r_A||ID(Bob)||K||t_{Bob}).$$

   The TA sends $y_1$ to Alice.

3. Alice decrypts $y_1$ using $K_{Alice}$. She verifies that it yielded the plaintext that is of the form $r_A||ID(Bob)||K||t_{Bob}$. At this point, Alice accepts $K$. She then sends $t_{Bob}$ to Bob.

4. Bob decrypts $t_{Bob}$ using $K_{Bob}$ to obtain $K$. Then Bob chooses a random number $r_B$ and computes $y_2 = e_K(r_B)$. He sends $y_2$ to Alice.

5. Alice decrypts $y_2$ using the session key $K$ to obtain $r_B$. She computes $y_3 = e_K(r_B - 1)$ and sends $y_3$ to Bob.

6. Bob decrypts $y_3$ using the session key $K$ and verifies that the plaintext is $r_B - 1$. At this point, Bob accepts $K$.

---

1Again, our discussion is based on *Cryptography: Theory and Practice*, 3rd Edition by Stinson. Section 9.6 of your book describes Kerberos in detail.
A summary of the main steps:

An attack:
Kerberos

Kerberos is a SKDS scheme that was developed at MIT in the late 1980’s and early 1990’s. It was partly based on the Needham-Schroeder SKDS. In Kerberos, time is incorporated into the protocol to prevent attacks like the one described above. Here is a simplified description of Kerberos, version 5.

1. Alice chooses a random number \( r_A \). She sends ID(Alice), ID(Bob) and \( r_A \) to the TA.

2. The TA chooses a random session key \( K \) and a validity period for \( K \) denoted by \( L \). It computes two values:

\[
    t_{Bob} = e_{K_{Bob}}(K||ID(Alice)||L),
\]

and

\[
    y_1 = e_{K_{Alice}}(r_A||ID(Bob)||K||L).
\]

The TA sends both \( t_{Bob} \) and \( y_1 \) to Alice.

3. Alice decrypts \( y_1 \) using \( K_{Alice} \) and verifies that it yielded a plaintext of the form \( r_A||ID(Bob)||K||L \). Alice then determines the current time, \( time \), and she computes

\[
    y_2 = e_K(ID(Alice)||time).
\]

Finally, Alice sends \( t_{Bob} \) and \( y_2 \) to Bob.

4. Bob decrypts \( t_{Bob} \) using his key \( K_{Bob} \) and verifies that it yielded a plaintext of the form \( K||ID(Alice)||L \). He also decrypts \( y_2 \) using the session key \( K \) to obtain \( time \). He then checks that \( time < L \). At this point, Bob accepts \( K \). Bob then computes

\[
    y_3 = e_K(time + 1)
\]

which he sends to Alice.

5. Alice decrypts \( y_3 \) using the session key \( K \) and verifies that the plaintext is \( time + 1 \). At this point, she accepts \( K \).

Discussion.