



PRINCIPLES IN PRACTICE

KERBEROS

Kerberos [RFC 1510; Neuman 1994] is an authentication service developed at MIT that uses symmetric key encryption techniques and a key distribution center. Although it is conceptually the same as the generic key distribution center (KDC) we describe in Section 8.5.1, its vocabulary is slightly different. Kerberos also contains several nice variations and extensions of the basic KDC mechanisms. Kerberos was designed to authenticate users accessing network servers and was initially targeted for use within a single administrative domain such as a campus or company. Thus, Kerberos is framed in the language of users who want to access network services (servers) using application-level network programs such as Telnet (for remote login) and NFS (for access to remote files), rather than human-to-human conversants who want to authenticate themselves to each other, as in our earlier examples. Nonetheless, the key (pun intended) underlying techniques remain the same.

The Kerberos authentication server (AS) plays the role of the KDC. The AS is the repository of not only the secret keys of all users (so that each user can communicate securely with the AS) but also information about which users have access privileges to which services on which network servers. When Alice wants to access a service on Bob (who we now think of as a server), the protocol closely follows our example in Figure 8.19.

1. Alice contacts the Kerberos AS, indicating that she wants to use Bob. All communication between Alice and the AS is encrypted using a secret key that is shared between Alice and the AS. In Kerberos, Alice first provides her name and password to her local host. Alice's local host and the AS then determine the one-time secret session key for encrypting communication between Alice and the AS.
2. The AS authenticates Alice, checks that she has access privileges to Bob, and generates a one-time symmetric session key, $R1$, for communication between Alice and Bob. The authentication server (in Kerberos parlance, now referred to as the Ticket Granting Server) sends Alice the value of $R1$, and also a ticket to Bob's services. The ticket contains Alice's name, the Alice–Bob session key, $R1$, and an expiration time, all encrypted using Bob's secret key (known by only Bob and the AS), as in Figure 8.19. Alice's ticket is valid only until its expiration time, and it will be rejected by Bob if presented after that time. For Kerberos V4, the maximum lifetime of a ticket is about 21 hours. In Kerberos V5, the lifetime must expire before the end of year 9999, a definite Y10K problem!
3. Alice then sends her ticket to Bob. She also sends along an $R1$ -encrypted timestamp that is used as a nonce. Bob decrypts the ticket using his secret key, obtains the session key, and decrypts the timestamp using the just-learned session key. Bob sends back the nonce to Alice, encrypted using $R1$, thus showing that Bob knows $R1$ and is live.

The most recent version of Kerberos (V5) provides support for multiple authentication servers, delegation of access rights, and renewable tickets. [Kaufman 1995] and [RFC 1510] provide ample details.

