Authentication of Unknown Entities on an Insecure Network of Untrusted Workstations

B. Clifford Neuman†, *Jennifer G. Steiner*

Project Athena
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

(bcn@june.cs.washington.edu, steiner@athena.mit.edu)

Project Athena provides computing resources for undergraduate education at MIT.[1] Over 750 computers are scattered across 30 subnets, and support more than 5,000 active users. Single user IBM RT/PCs and DEC VaxStation IIs running versions of the Unix operating system access servers (mostly Vax 11/750s) across the network. Workstations are publicly and privately owned. In both cases the user has complete control over the computer and can easily gain superuser status. Because of this, workstations cannot be trusted to accurately identify their users. The network can't be considered secure either. Users can listen to network traffic as well as generate traffic with forged addresses. Servers are scattered across campus. It is possible that users might be able to physically compromise the security of some of the them.

A method was needed to authenticate users wishing to access network services such as file storage, electronic mail, remote login, and printing. The method had to be secure in the given environment, but not unduly cumbersome for the user. Ideally, the system would appear to the user as if only a single system were being used. Any solution chosen had to scale well. Additionally, compromise of any of the servers could not affect the security of the others.

The approach taken is based on a cryptographic protocol by Needham and Schroeder.[2] An authentication server known as *Kerberos*[3, 4] runs on a trusted computer. Kerberos knows the passwords (encryption keys) for each user under its authority. It also shares a key with each server. When a program running on a workstation (e.g. *rlogin)* wishes to prove the identity of its user to a given network server (e.g. *rlogind)*, it contacts Kerberos and asks for a *ticket* for that server. The ticket is returned to the workstation encrypted in the server's key, and then again in the user's key. The user's password is used to decrypt the ticket which can then be passed to the server to prove the user's identity.

In addition to the ticket, Kerberos generates and returns to the user a temporary encryption key, or *session key*. This, like the ticket, is sealed in the user's password. A copy of the session key is also enclosed in the server ticket. Once the server decrypts the ticket with its key, both the user and server know the session key, which can be used to encrypt further communication between them. In this way, the Kerberos server also acts as a key distribution center.

A ticket can be reused, but additional information passed to the server along with the ticket prevents replays by an imposter. The initial ticket obtained is for a *ticket-granting service* which can be used to obtain tickets for other services. In this way, the user only has to enter a password once per login session. Tickets have a finite lifetime, and an attacker who manages to steal tickets from a user can use them for only a short time (relative to the life of the user's password), and only from a particular network address.

Under the Kerberos model, the world is divided into separate domains of authentication authority,

† Author's present address: B. Clifford Neuman, Department of Computer Science(FR-35), University of Washington, Seattle, Washington 98195.

called *realms*, each with its own Kerberos server. Principals registered in one realm can easily authenticate themselves to servers in other realms. This is accomplished through ticket-granting servers which are registered in multiple realms.

Kerberos is implemented as a server that runs on a secure machine, and a set of libraries that is used by client applications and services. The initial implementation uses DES for encryption, but encryption is supported in a separate module that is easily replaced.

Kerberos has been in use at MIT for two years, and is currently in beta test at 18 sites across the country. At MIT, Kerberos supports more than 8,000 entities (users and servers) in three different realms. It is used for authentication in rsh, rcp, rlogin, Sun's Network File System, mail, bulletin boards, notification and administrative applications. In summary, Kerberos allows users to authenticate themselves to network services without entering a password at every request, and without relying on less secure methods, such as the host-authenticated *.rhost* mechanism.