Kerberos

COMP620

Kerberos

- An encryption and authentication protocol developed at MIT in the 1980s
- Used in Microsoft Windows Active Directory
- Uses symmetric key encryption
  - First versions used DES
  - AES is now used
- Kerberos provides authentication and secure access of application servers (i.e. file servers, print servers, etc.)

Servers

Kerberos requires two servers for security

- Authentication Server (AS) that validates a user
- Ticket Granting Server (TGS) that provides access to application servers if the user is allowed
- Frequently both server functions are housed in the same machine

Keys

- Every user has a secret symmetric encryption **key** known only to them and the Authentication server. The key is created from their password.
- There is an encryption **key** known only to the AS and TGS
- Each application server has an encryption **key** known only to the server and TGS
- New **keys** are created for connection to servers

Time Stamps

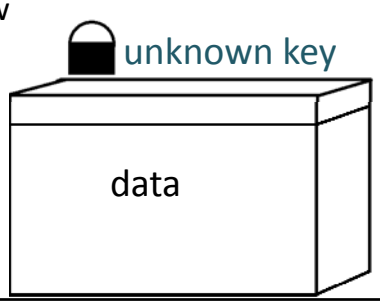
- All messages contain the current time to prevent replay attacks
- Tickets are only good for a specified length of time
- *Time stamps are not shown in the following diagrams*

Nonce

- A **nonce** *(abbreviation of number used once)* is a random number
- The nonce of a message is encrypted and included in the reply
- Nonces are included in messages and replies to ensure that the reply is fresh and applies to the recent request
- Included to prevent replay attacks

Tickets

- A ticket contains information, such as a new encryption key or user ID
- Tickets are encrypted. The client will receive tickets encrypted by keys it does and/or does not know



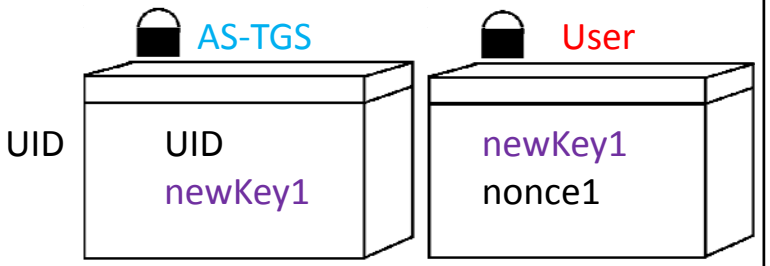
Overview

- A client is authenticated by the AS
  - This usually happens only once a day or
- A client asks a TGS for permission to use an application server
  - A request is required for each server

Client Login to AS

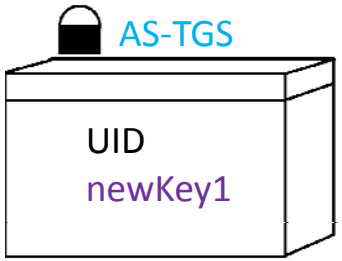
- A client first sends its userid (UID) and nonce1 to the authentication server
- Everything is sent in plaintext

AS to Client Reply



The client decrypts a portion of the message to get the new key 1. It checks to make sure the nonce1 is the same one it sent.

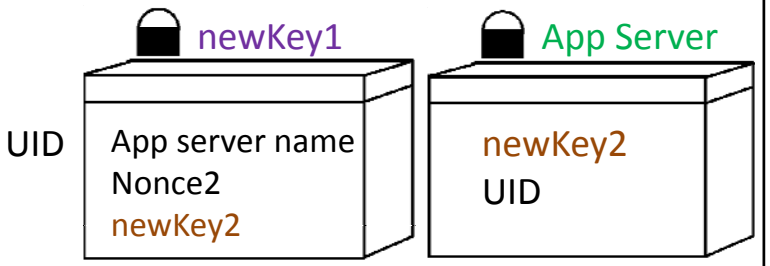
Client to TGS Request



App server name
Nonce2
UID

TGS trusts data encrypted by AS. The TGS checks if the client is allowed to access the app server.

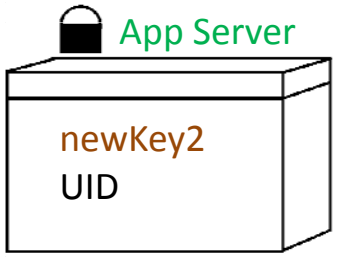
TGS to Client Reply



The client decrypts a portion of the message to get the new key 2 for the app server. It checks to make sure the nonce2 is correct.

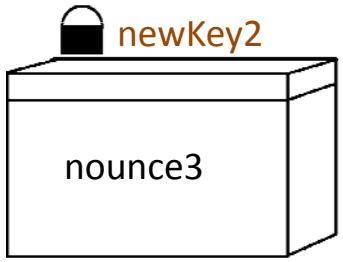
Client Request to App Server

UID Nonce3



The App Server trusts the request because it contains a ticket encrypted using the key known only to itself and the TGS

App Server Reply to Client



All further communication with the App Server is encrypted using newKey2