KERBEROS PROTOCOL

with 2-Factor Authentication

June, 2020

Nava Lorenzo,
Riva Lorenzo,
Vignati Edoardo

Università degli Studi di Milano

Kerberos Protocol with 2-Factor Authentication

Introduction

The idea

Our idea was to implement the **Kerberos Protocol** by adding two new features:

- **2-Factor Authentication** via TOTP,
- the possibility of registering new users to the Authentication Server using **asimmetric encryption**.

Implementation

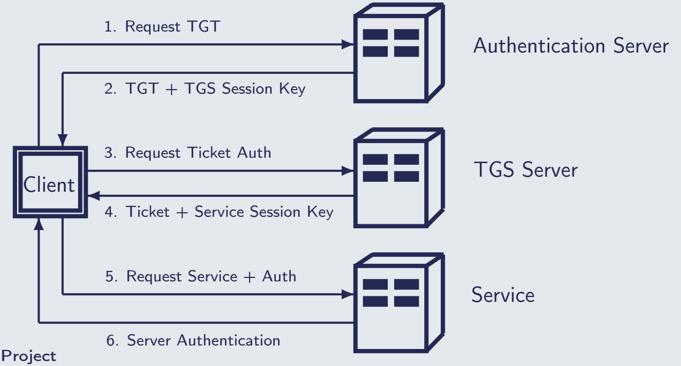
The project is written in Java and uses Jetty and Jersey which provide a REST server implementation.
Every message is represented by a dedicated Java class which is serialized/deserialized with the Gson library.
The tokens are encrypted using the default Cipher utilities from Java standard library.

The Protocol

Kerberos

Kerberos is an authentication service. It works on the basis of tickets allowing the devices to prove their identity with each other in a secure manner, communicating over a non-secure network.

Schema



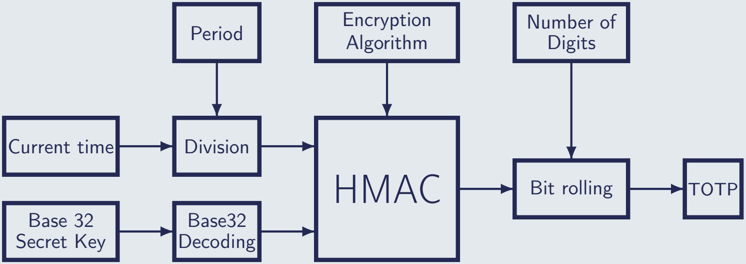
2-Factor Authentication: OTP (1)

UNIVERSITÀ DEGLI STUDI DI MILANO

RFC 4226 - HOTP
RFC 6238 - TOTP

Structure of a Time-Based One-Time Password

- Client
  - QRcode viewer
  - Authentication app (eg. Google Authenticator)
- Server



2-Factor Authentication: OTP (2)

Kerberos + OTP

Our implementation integrates the OTP into Kerberos maintaining its design.

- The OTP key is generated by the server and provided to the client during the registration step.
- During the authentication, the server creates the TGS token adding the OTP code too.
- The client knows the OTP code too and sends it to the ticket granting server.
- The TGS compares the OTP codes.

Registration: asymmetric cryptography

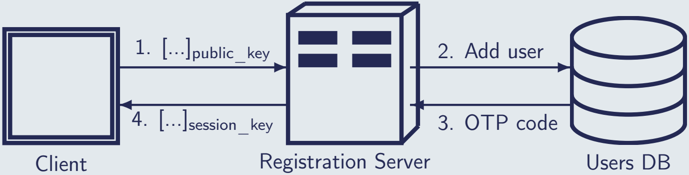
UNIVERSITÀ DEGLI STUDI DI MILANO

Message exchange

The asymmetric encryption is used during the registration step

- Client → Server: $[user||password||session\_key]_{public\_key}$
- Client ← Server: $[OTP\_secret\_key]_{session\_key}$

Schema



Thank you!