# UNIVERSITÀ DI PISA
# FACOLTÀ DI INGEGNERIA

**Project of**

# SECURITY IN NETWORKED COMPUTING SYSTEM

**Fabio Greco, Pietro Piscione**

**A.Y. 2014/15**

# Index

# 1. Intro

## 1.a Objective

Consider a distributed client-server application where each client 'A' shares a long-term secret with the server 'B'.
Assuming you are in a situation of mutual-trust, you specify, analyze, projects and finally implements
a cryptographic protocol that meets the following requirements:

at the end of the execution of the protocol, it establishes a session key between 'A' and 'B'; and always at the end of the execution of the protocol, the client 'A' believes that the server 'B' has the session key and vice versa;
the session key is generated by the server 'B'.

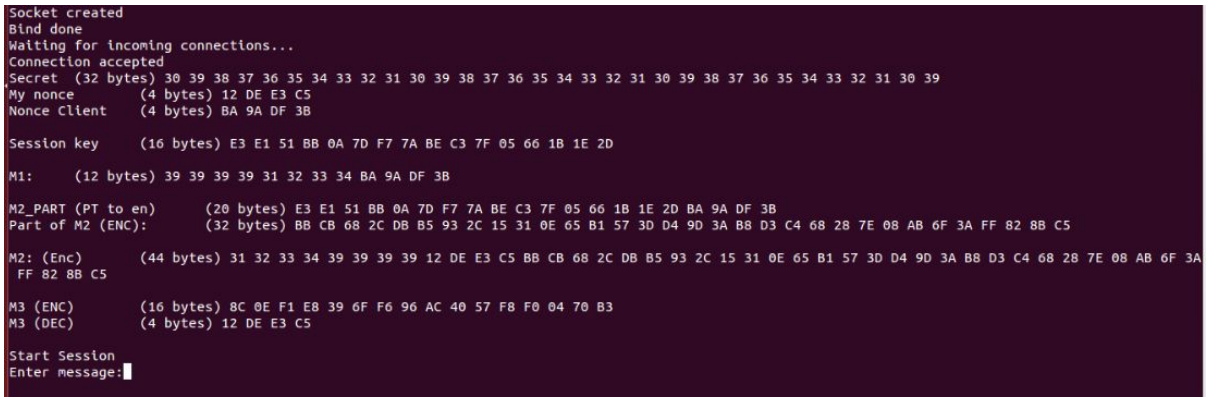
The protocol specifications has to clearly highlight the assumptions under which the protocol is working properly.
The implementation has to include the realization of a prototype in which the server and the client exchange of the ciphertext with the session key.

## 1.b Acronyms & glossary

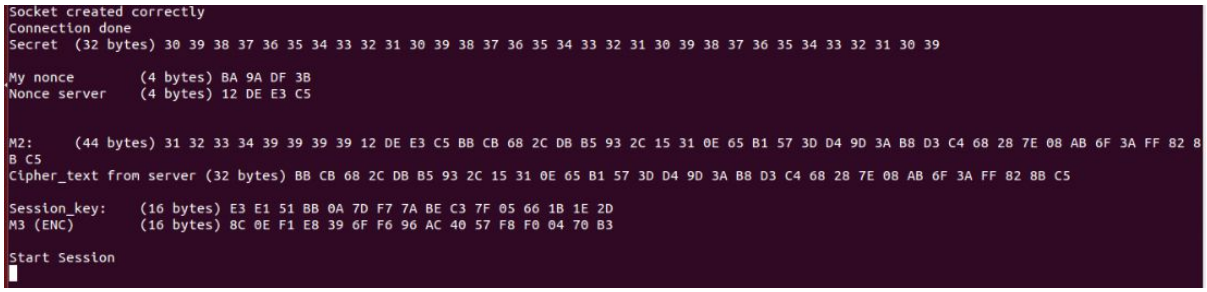
**sk :** Shared Secret between client A and server B;
**Na:** nonce of client A;
**Nb:** nonce of server B;
**IdA:** identificator of client A;
**IdB:** identificator of server B;
**$K_{AB}$:** Session Key.

# 1.c Preview of finished project

server:




client:



# 2. Implementation

This project is divided into three phases.

The first one ( exchange of a key by hand ) that starts before the use of the application.

The second phase implements the session key establishment and after the key confermation between client and server.

The last one implements the encrypted session through the session key.

In order to guarantee the integrity of the exchanged bytes was implemented the hash function. In this specific project the hash type used is “sha256”**.**

Anyway we made few assumptions for the first phase in order to properly deploy the protocol.

# 3. Assumption

## 3.a Not tech

- The first phases, as already said, involves the exchange of a key, defined as shared secret ( **sk** ), which was trading by hand between client and server;

- We suppose that the shared secret is stored in a password protected file (sk_file);

- When the client connects to the server, it is already online;

- The shared secret sk is stored in a file encrypted by a local password;

- The local password is known a priori from the client and the server.

## 3.b Tech

- Nonce size : 4 Byte (32 bit) ;
- IdA and IdB size : 4 Byte (32 bit);
- Shared Secret size : 32 byte (256 bit)
  The shared secret is bigger than the session key because the Shared Secret have a life time bigger than the other one;
- Session Key size : 16 byte (128 bit).

# 4. The protocol and its analysis

Key session estabilishment is following illustrated:

**M1**: A → B IdA, IdB, Na
**M2**: B → A IdB, IdA, Nb, $\{ K_{AB} , Na \}_{\mathbf{sk}}$
**M3**: A → B $\{ K_{AB} , Nb\}_{\mathbf{K}_{AB}}$

## 4.a Analysis:

### M1

Here starts the comunication, so the client **A** sends to the server **B** his nonce **Na** that the server will use to evaluate the freshness of this message when it will be received, the identificator **IdB** that is the Id that identify the server in a univocal way and the **IdA** that do the same for the identification of the sender, the client A.

### M2

During this second message the shared secret key is used for the first time to encrypt the session key $\mathbf{K}_{AB}$ that the server have generated.
This $\mathbf{K}_{AB}$ is also linked ahead to the nonce **Na** that **A** had previously sent to the server, thus both the session key and the nonce will be encrypted with **AES 256bit CBC**, while the rest of the message will be in clear.

The part of the plaintext message will include the identifiers of the server ( **IdB** ) and of the client ( **IdA** ) and the server nonce ( **Nb** ), added in this message in order that the client can check the freshness of communication too.

# M3

The client receives M2 and decrypts it with the Sk in order to get the session key.
Before that, the client checks the message freshness with the nonce comparison.

Now client A is ready to sends back the Nb to the server B in order to conclude the second phase with the key confirmation.
This communication is encrypted through **AES 128 CBC**.

# 5. BAN logic analysis

Now, let us analyze the protocol through the BAN logic rules.
First of all, let us derive the idealized protocol from the real one:

**M1**: $A \rightarrow B$ IdA, IdB, Na
**M2**: $B \rightarrow A$ IdB, IdA, Nb, $\langle A \leftarrow K_{AB} \rightarrow B , Na \rangle_{sk}$
**M3**: $A \rightarrow B$ $\{ A \leftarrow K_{AB} \rightarrow B, Nb \}_{K_{AB}}$

## 5.a Hypothesis

The hypothesis are the following:

- $A \overset{SK}{\leftrightarrows} B$

  Long term shared secret is shared between A and B ;

- $A \mid\equiv \#(Na)$ ;
- $B \mid\equiv \#(Nb)$ ;
- $B \mid\equiv \#(Na)$ ;
- $A \mid\equiv \#(Nb)$ ;
- $B \Rightarrow A \leftarrow K_{AB} \rightarrow B$

## 5.b Objectives

The objectives that we want to reach at the end of the protocol are the following:

Key Authentication

- $A \mid\equiv A \leftarrow K_{AB} \rightarrow B$
- $B \mid\equiv A \leftarrow K_{AB} \rightarrow B$

Key Confirmation

- $A \mid\equiv B \mid\equiv A \leftarrow K_{AB} \rightarrow B$
- $B \mid\equiv A \mid\equiv A \leftarrow K_{AB} \rightarrow B$

Key Freshness

- $A \mid\equiv \#(A \leftarrow K_{AB} \rightarrow B)$
- $B \mid\equiv \#(A \leftarrow K_{AB} \rightarrow B)$

## 5.c Postulates used

- $$\frac{B \mid\equiv A \overset{K_{AB}}{\leftrightarrow} B \;,\; B \triangleleft \{ X \}_{K_{AB}}}{B \mid\equiv A \mid\sim X}$$

If SK is a shared key between B and A, and B sees a message encrypted by SK containing X (and B didn't send that message), then B believes that X was sent by A.

- $$\frac{B \mid\equiv \overset{K_{ab}}{\mapsto} A \;,\; B \triangleleft \{ X \}_{K_{ab}}}{B \mid\equiv A \mid\sim X}$$

If $K_{AB}$ is the session key used from A, and B sees a message signed by $K_{AB}$ containing X, then B believes that X was sent by A.

- $$\frac{B \mid\equiv A \overset{SK}{=\!\!=} B \;,\; B \triangleleft \langle X \rangle_{SK}}{B \mid\equiv A \mid\sim X}$$

If SK is a shared secrete between B and A, and B sees a message where SK is combined with X (and B didn't send the message), then B believes that X was sent by A.

## 5.d Proof

### After M2

$$A \triangleleft \langle K_{AB}, Na \rangle_{SK}, A \mid\equiv A \overset{SK}{\leftrightharpoons} B, B \mid\equiv \#(K_{AB})$$

---

$$A \mid\equiv B \mid\sim \langle K_{AB}, Na \rangle_{SK}, A \mid\equiv \#(Na), A \mid\equiv \#(K_{AB})$$

---

$$A \mid\equiv B \mid\equiv K_{AB}, A \mid\equiv B \Rightarrow K_{AB}$$

---

$$A \mid\equiv K_{AB}$$

### After M3

$$B \triangleleft \{ Nb, K_{AB} \}_{K_{AB}}, B \mid\equiv A \overset{K_{AB}}{\leftrightarrow} B,$$

---

$$B \mid\equiv A \mid\sim (Nb, K_{AB}), B \mid\equiv \#(Nb)$$

---

$$B \mid\equiv A \mid\equiv K_{AB},$$

# 6. Reference

About theory:

http://www.iet.unipi.it/g.dini/Teaching/sncs/lectures/index.html

About code:

http://www.iet.unipi.it/p.perazzo/teaching/