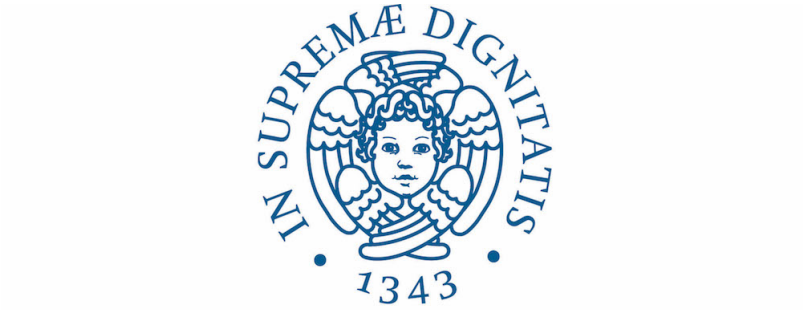

UNIVERSITÀ DI PISA

MSc in Computer Engineering
Cybersecurity
A.A. 2018/2019

Project Report

Secure File Transfer

Alessio Ercolani
Federico Garzelli

Contents

1 Design Choices

At the end of the key establishment protocol, described in section 2, client and server share two symmetric session keys: k_{conf} and k_{auth}, used to protect respectively confidentiality and authenticity. These keys are obtained starting from $k = g^{cs} \mod p$, which is computed with the Diffie-Hellman protocol, by applying SHA-512 algorithm to it and then dividing the digest into two parts with the same length: k_{conf} is composed by the high-order bits, while k_{auth} is composed by the low-order bits. k_{conf} and k_{auth} are deleted at the end of the session.

Once the session keys are established, we use AES-256 in cipher block chaining (CBC) mode to achieve confidential communication and HMAC based on SHA-256 to authenticate messages between client and server. The initialization vector is randomly generated before sending each packet and is inserted in the message.

During the key establishment protocol, client and server sign part of their messages with digital signature based on SHA-256. The server receives client's certificate during this protocol, while server's certificate is pre-installed in client's file system. The parameters p and g are hard-coded in both client's and server's installations, just to avoid the computations necessary to generate a 2048 bit long prime number.

The communication is also protected against replay attacks. During the key establishment protocol fresh quantities are generated, while then the header of the messages contains the packet sequence number. Both client and server keep two counters: one for the messages they sent, one for the messages they receives. They put the former in each message they send and check if it corresponds to the latter at each message they receive.

2 Key Establishment Protocol

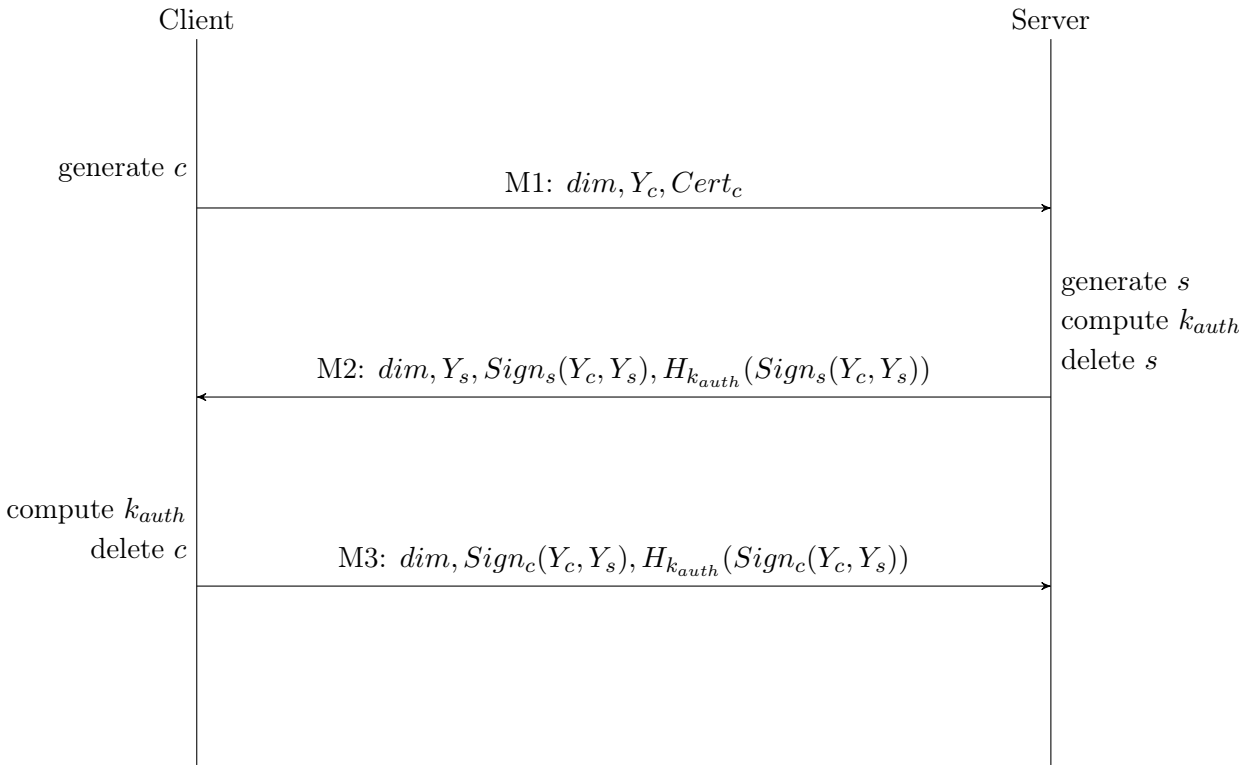

Figure 1: Key establishment protocol

3 BAN Logic Proof

In this section, k_{auth} is called k to avoid weighing down the notation.

3.1 Idealized Protocol

- M1 $C \to S:\ Y_c,\ \{\overset{K_c}{\longmapsto} C\}_{K_{ca}^{-1}}$
- M2 $S \to C:\ Y_s,\ \{(Y_c, Y_s)\}_{K_s^{-1}},\ \langle (Y_c, Y_s), C \overset{k}{\leftrightarrow} S\rangle_k$
- M3 $C \to S:\ \{(Y_c,\ Y_s)\}_{K_c^{-1}},\ \langle (Y_c, Y_s), C \overset{k}{\leftrightarrow} S\rangle_k$

3.2 Objectives

Key authentication

$$C \mid\equiv C \overset{k}{\leftrightarrow} S \tag{1}$$

$$S \mid\equiv C \overset{k}{\leftrightarrow} S \tag{2}$$

Key confirmation

$$C \mid\equiv S \mid\equiv C \overset{k}{\leftrightarrow} S \tag{3}$$

$$S \mid\equiv C \mid\equiv C \overset{k}{\leftrightarrow} S \tag{4}$$

Key freshness

$$C \mid\equiv \#(C \overset{k}{\leftrightarrow} S) \tag{5}$$

$$S \mid\equiv \#(C \overset{k}{\leftrightarrow} S) \tag{6}$$

3.3 Assumptions

Keys

- $C \mid\equiv \overset{K_c}{\longmapsto} C$
- $S \mid\equiv \overset{K_s}{\longmapsto} S$
- $C \mid\equiv \overset{K_{ca}}{\longmapsto} CA$
- $S \mid\equiv \overset{K_{ca}}{\longmapsto} CA$

Freshness

- $C \mid\equiv \#(c)$
- $S \mid\equiv \#(s)$

Trust

- $C \mid\equiv\ CA \Rightarrow \overset{K_s}{\longmapsto} S$
- $S \mid\equiv\ CA \Rightarrow \overset{K_c}{\longmapsto} C$

3.4 Proof

Before M1

C generates a random fresh c, computes $Y_c = g^c \mod p$, thus, with freshness promotion:

$$\frac{C \mid\equiv \#(c)}{C \mid\equiv \#(Y_c)}$$

After M1

$S \triangleleft Y_c$, generates a random fresh s, computes $k = (Y_c)^s \mod p$, thus, with freshness promotion:

$$\frac{S \mid\equiv \#(s)}{S \mid\equiv \#(k)}$$

S computes $Y_s = g^s \mod p$, thus, with freshness promotion:

$$\frac{S \mid\equiv \#(s)}{S \mid\equiv \#(Y_s)}$$

With message meaning rule:

$$\frac{S \triangleleft \{\stackrel{K_c}{\longmapsto} C\}_{K_{ca}^{-1}},\ S \mid\equiv \stackrel{K_{ca}}{\longmapsto} CA}{S \mid\equiv CA \mid\sim \stackrel{K_c}{\longmapsto} C}$$

Moreover $S \triangleleft \stackrel{K_c}{\longmapsto} C$. S verifies CA's CRL and validity period of C's certificate, then $S \mid\equiv CA \mid\equiv \stackrel{K_c}{\longmapsto} C$. With jurisdiction rule:

$$\frac{S \mid\equiv CA \mid\equiv \stackrel{K_c}{\longmapsto} C,\ S \mid\equiv CA \Rightarrow \stackrel{K_c}{\longmapsto} C}{S \mid\equiv \stackrel{K_c}{\longmapsto} C}$$

After M2

C reads S's certificate, which is installed in their file system, to get S's public key: $C \triangleleft \{\stackrel{K_s}{\longmapsto} S\}_{K_{ca}^{-1}}$; thus, with message meaning rule:

$$\frac{C \triangleleft \{\stackrel{K_s}{\longmapsto} S\}_{K_{ca}^{-1}},\ C \mid\equiv \stackrel{K_{ca}}{\longmapsto} CA}{C \mid\equiv CA \mid\sim \stackrel{K_s}{\longmapsto} S}$$

Moreover $C \triangleleft \stackrel{K_s}{\longmapsto} S$. C verifies CA's CRL and validity period of S's certificate, then $C \mid\equiv CA \mid\equiv \stackrel{K_s}{\longmapsto} S$. With jurisdiction rule:

$$\frac{C \mid\equiv CA \mid\equiv \stackrel{K_s}{\longmapsto} S,\ C \mid\equiv CA \Rightarrow \stackrel{K_s}{\longmapsto} S}{C \mid\equiv \stackrel{K_s}{\longmapsto} S}$$

With message meaning rule:

$$\frac{C \triangleleft \{Y_c, Y_s\}_{K_s^{-1}},\ C \mid\equiv \stackrel{K_s}{\longmapsto} S}{C \mid\equiv S \mid\sim (Y_c, Y_s)}$$

With freshness promotion:

$$\frac{C \mid\equiv \#(Y_c)}{C \mid\equiv \#((Y_c, Y_s))}$$

With nonce verification rule:

$$\frac{C \mid\equiv S \mid\sim (Y_c, Y_s),\ C \mid\equiv \#((Y_c, Y_s))}{C \mid\equiv S \mid\equiv (Y_c, Y_s)}$$

Now $C \triangleleft Y_s$, computes $k = (Y_s)^c \mod p$, thus, with freshness promotion:

$$\frac{C \mid\equiv \#(c)}{C \mid\equiv \#(k)}$$

With $\overset{k}{\leftrightarrow}$-introduction rule:

$$\frac{C \mid\equiv \#(k),\ C \mid\equiv S \mid\equiv (Y_c, Y_s)}{C \mid\equiv C \overset{k}{\leftrightarrow} S}$$

which is objective (1). We can also state that $C \mid\equiv \#(C \overset{k}{\leftrightarrow} S)$, which is objective (5). With message meaning rule:

$$\frac{C \triangleleft \langle C \overset{k}{\leftrightarrow} S \rangle_k,\ C \mid\equiv C \overset{k}{\rightleftharpoons} S}{C \mid\equiv S \mid\sim C \overset{k}{\leftrightarrow} S}$$

With nonce verification rule:

$$\frac{C \mid\equiv \#(C \overset{k}{\leftrightarrow} S),\ C \mid\equiv S \mid\sim C \overset{k}{\leftrightarrow} S}{C \mid\equiv S \mid\equiv C \overset{k}{\leftrightarrow} S}$$

which is objective (3).

After M3

With message meaning rule:

$$\frac{S \triangleleft \{Y_c, Y_s\}_{K_c^{-1}},\ S \mid\equiv \overset{K_c}{\longmapsto} C}{S \mid\equiv C \mid\sim (Y_c, Y_s)}$$

With freshness promotion:

$$\frac{S \mid\equiv \#(Y_s)}{S \mid\equiv \#((Y_c, Y_s))}$$

With nonce verification rule:

$$\frac{S \mid\equiv C \mid\sim (Y_c, Y_s),\ S \mid\equiv \#((Y_c, Y_s))}{S \mid\equiv C \mid\equiv (Y_c, Y_s)}$$

With $\overset{k}{\leftrightarrow}$-introduction rule:

$$\frac{S \mid\equiv \#(k),\ S \mid\equiv C \mid\equiv (Y_c, Y_s)}{S \mid\equiv C \overset{k}{\leftrightarrow} S}$$

which is objective (2). Since $S \mid\equiv \#(k)$, we can state that $S \mid\equiv \#(C \overset{k}{\leftrightarrow} S)$, which is objective (6). With message meaning rule:

$$\frac{S \triangleleft \langle C \overset{k}{\leftrightarrow} S \rangle_k,\ S \mid\equiv C \overset{k}{\rightleftharpoons} S}{S \mid\equiv C \mid\sim C \overset{k}{\leftrightarrow} S}$$

With nonce verification rule:

$$\frac{S \mid\equiv \#(C \overset{k}{\leftrightarrow} S),\ S \mid\equiv C \mid\sim C \overset{k}{\leftrightarrow} S}{S \mid\equiv C \mid\equiv C \overset{k}{\leftrightarrow} S}$$

which is objective (4).

4 Message Format

After the end of the key establishment protocol, client and server exchange structured messages. Messages start with a fixed-size header, then have the payload and an HMAC field. Figure 2 shows header format: the packet sequence number, the length of the payload, a code to identify the type of the message and an HMAC of the other header fields. Figure 3 shows payload format: the initialization vector and the ciphertext.

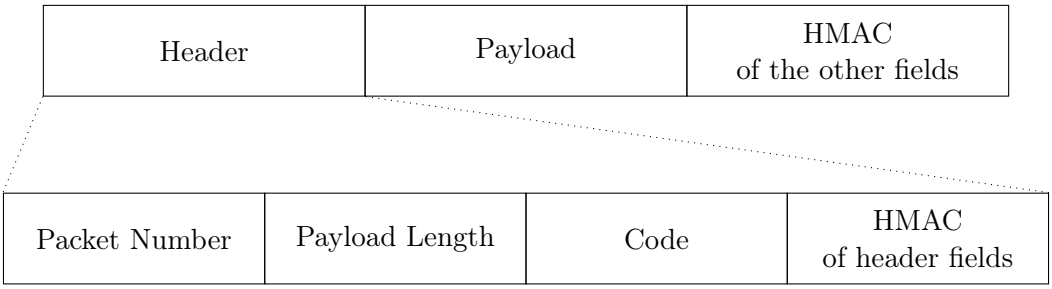

Figure 2: Header format

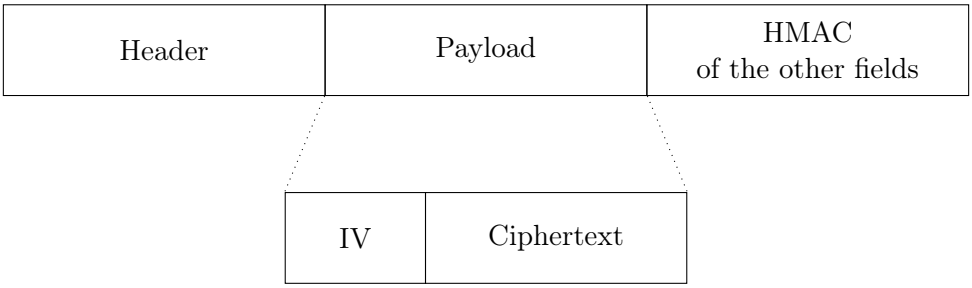

Figure 3: Payload format