

Projet de sécurité des protocoles

PROTOCOL v.2

Ambroise BAUDOT, Marceau DIDA, Louise POUNT

January 31, 2020

Overview Let S be a honest server. What's more, hosts (clients) are able to generate fresh keys (here, a key K). The protocol can be described as follows:

1. $A \rightarrow S : A, \{ \langle B, K \rangle \}_{AS}$
2. $S \rightarrow B : \{ \langle A, K \rangle \}_{BS}$
3. $B \rightarrow A : \{ \{N_B\}_K \}_{pk_A}$
4. $A \rightarrow B : \{N_B\}_{pk_B}$
5. $A \rightarrow B : \{ \{N_A\}_K \}_{pk_B}$
6. $B \rightarrow A : \{N_A\}_{pk_A}$

Initialization Let a symmetric key be shared between the server S and the host A , and the same with S and the host B .

Generated data during the process A fresh symmetric key is generated by the initiator host, for the 1st message, *i.e.* for each session beginning. Each client generates a temporary fresh nonce too (lifespan limited to the two last messages).

Protocol description A initiates the protocol generating a symmetric key K , which will theoretically be the symmetric key shared between A and B . Then A sends the identity of the target host, in a pair with the key K , to the server S . This message is encrypted with the common key to A and B . Then, the server unwraps the message to forward the key to B (present in the input message), in pair with the identity of the initiator. The aim of these two messages is to preserve the **secret** of the key K .

Then, messages (3, 4) (respectively (5, 6)) are a challenge sent by B (resp. A) to ensure that the other host is really B (respectively A). This challenge is on one hand a "proof of knowledge" (A and B must know K , and the secret is preserved thanks to messages (1, 2)) and on the other hand an authentication thanks to the public-key encryption. That is to satisfy the **authentication** property.

Safety queries

- **Secret:** The fresh key K generated must be known only by A and B .
- **Authentication:** Hosts A and B must have been mutually authenticated.

Cost Let C be the cost of the protocol, and c_i the cost of the i^{th} message.

$$\begin{aligned}C &= 2 * (c_1 + c_3 + c_4) + 1 \quad (+1 \text{ pour le message 1}) \\C &= 2 * ((10 + 50 + 1 + 1) + (1 + 10 + 1) + (1 + 1)) + 1 \\C &= 153\end{aligned}$$