Security Protocols and Verification

Design and Analysis of Cryptographic Protocols

Garance Frolla
Ely Marthouret
Ewan Decima

Team: **ASKO OM8464A2**

September / November 2025

Contents

1 Initial Knowledge **2**

2 Ely the big frog **2**

3 Protocol Description **2**

3.1 Messages cost and description 2
3.2 Value Generated 3
3.3 Security Properties 3

1 Initial Knowledge

At the beginning of the protocol, agents A and B are assumed to know all public keys, especially each other's. Both A and B also know their respective symmetric keys (K_{AS} and K_{BS}) shared with an honest server S and there clocks are synchronised on the S one. Furthermore, each participant is aware of the chosen protocol and the following specifications.

2 Ely the big frog

1. $A \rightarrow B : \{|\langle A, N_A \rangle|\}_{K_{AB}}$
2. $A \rightarrow S : \{|\langle B, \tau, \lambda, K_{AB} \rangle|\}_{K_{AS}}$
3. $S \rightarrow B : \{|\langle A, \tau, \lambda, K_{AB} \rangle|\}_{K_{BS}}$
4. $B \rightarrow A : \{|N_A + 1|\}_{K_{AB}}$

3 Protocol Description

3.1 Messages cost and description

This protocol begins with entity A generating a nonce, denoted as N_A. A then encrypts her identity together with the nonce using the freshly generated session key K_{AB}. The transmitted data is structured as follows: $\{|\langle A, N_A \rangle|\}_{K_{AB}}$. This message costs **63**.

After sending the first message, A sends to the honest and trusted server S, using the shared key K_{AS}, the identity of B, a timestamp τ, a lifetime period λ to confirm the key and the session key K_{AB}. The transmitted data is structured as follows: $\{|\langle B, \tau, \lambda, K_{AB} \rangle|\}_{K_{AS}}$. This message costs **166**.

First, S checks whether $t \geq \tau + \lambda$, where t denotes the time at which S receives the message from A. If this condition holds, S aborts. Otherwise, using the shared key K_{BS}, S sends to B essentially the same message as before, except that A is replaced with B. The transmitted data is structured as follows: $\{|\langle A, \tau, \lambda, K_{AB} \rangle|\}_{K_{BS}}$. This message costs: **166**.

B receives the message $\{|\langle A, \tau, \lambda, K_{AB}\rangle|\}_{K_{BS}}$ and obtains the session key K_{AB}. He also learns the validity period λ, starting from time τ, during which A will accept his response. This measure provides protection against ticket theft. Indeed, even if an attacker manages to intercept a ticket, they will not be able to use it after its expiration. When B receives this message at time t, if $t \geq \tau + \lambda$, then B aborts.

Otherwise, B respond to the first message of A, he can decrypt $\{|\langle A, N_A\rangle|\}_{K_{AB}}$ with the session key. Key confirmation lies in the fact that B sends back $N_A + 1$ to A. In this way, A knows that B has successfully retrieved the key. This allows combining key confirmation with the challenge–response mechanism for the authentication of B with respect to A. The transmitted data is structured as follows: $\{|N_A + 1|\}_{K_{AB}}$. This message costs **12**.

At the end, when A receives the last message from B at time t, she checks whether $t \geq \tau + \lambda$. If this condition holds, A rejects the message and aborts. Otherwise, the key exchange protocol succeeds.

The total cost is: **409**.

3.2 Value Generated

- N_A is a nonce generated by A.
- K_{AB} is a *perfectly* random symmetric key generated by A.
- τ is a timestamp generated by A based on the clock synchronized by S.
- λ is a duration generated by A, small enough to be secure but long enough to establish a communication.

3.3 Security Properties

In addition to the security properties specified in the project description, we have:

- *Freshness*: K_{AB} is freshly generated by A, independently of B and of the current time. Each new session results in a newly generated key K_{AB}.

- *Authentification*: When B receives the first message $\{|\langle A, N_A \rangle|\}_{K_{AB}}$, he can't be sure that A sends it to him. But after receiving $\{|\langle A, \tau, \lambda, K_{AB} \rangle|\}_{K_{BS}}$ from S he can be sure of the origin of the first message. Indeed, A uses S to authenticate to B. Since the message containing A's identity is encrypted with K_{BS}, B can be sure that A initiated the conversation.

- *Key confirmation*: After receiving the third message, B obtains the session key K_{AB} from S. To confirm possession of this key, B sends back to A the value $N_A + 1$, encrypted under K_{AB}. This proves to A that B has indeed obtained the correct key.

- *Secrecy*: Nobody except for agents A and B know the session key K_{AB}. Indeed, all messages transmitted over the network are encrypted using symmetric encryption.

- *Integrity*: Each message is encrypted with a symmetric key, ensuring that only an entity possessing the corresponding key (K_{AB}, K_{AS}, or K_{BS}) can modify its contents.