Security Protocols and Verification

Design and Analysis of Cryptographic Protocols

Garance Frolla
Ely Marthouret
Ewan Decima

September / November 2025

1 Assumptions

We assume that at the beginning of the protocol agents A and B know the public key $K_{pub(C)}$ of any agent C. Moreover, we assume that all agents C shared a symmetric key with a trusted server S, named K_{CS}. N_a is a nonce generated by A, and N_b is a nonce generated by B. τ and λ respectively denote a timestamp and a lifetime. A generates K_{AB} with perfect randomness at each session. .

2 Protocol: First Attempt

1. $A \to S : \{|A, B, N_a, K_{AB}|\}_{K_{AS}}$
2. $S \to A : \left\{ \left| S, A, N_a + 1, \{|A, B, K_{AB}, N_a, \tau, \lambda|\}_{K_{BS}} \right| \right\}_{K_{AS}}$
3. $A \to B : \{|A, B, K_{AB}, N_a, \tau, \lambda, |\}_{K_{BS}}$
4. $B \to S : \{|B, A, N_b|\}_{K_{BS}}$
5. $S \to B : \{|S, B, N_b + 1, h(K_{AB}, N_b)|\}_{K_{BS}}$
6. $B \to A : \{B, A, N_a + 1, h(K_{AB}, N_b)\}_{pub(A)}$
7. $A \to B : \{A, B, N_b + 1, h(K_{AB}, N_a)\}_{pub(B)}$

3 Fourth Attempt

1. $A \rightarrow S : \{|A, B, N_a, K_{AB}|\}_{K_{AS}}$
2. $S \rightarrow A : \left\{ \left| A, N_a + 1, \{|A, B, N_a, \tau, \lambda, K_{AB}|\}_{K_{BS}} \right| \right\}_{K_{AS}}$
3. $A \rightarrow B : \{|A, B, N_a, \tau, \lambda, K_{AB}|\}_{K_{BS}}$
4. $B \rightarrow A : \{B, A, N_a + 1, h(K_{AB}, N_a)\}_{pub(A)}$

4 Protocol: Pub and Priv

1. $A \rightarrow B : \left\{ \{A, N_A, K_{AB}\}_{priv(A)} \right\}_{pub(B)}$
2. $B \rightarrow A : \left\{ \{B, N_A + 1, h(K_{AB})\}_{priv(B)} \right\}_{pub(A)}$

This *perfect* protocol does not conform to the requirement equations.

5 Protocol: pub + Kab

1. $A \rightarrow B : \left\{ \{A, N_A, K_{AB}\}_{pub(B)} \right.$
2. $B \rightarrow A : \left\{ \{B, N_A, N_B\}_{pub(A)} \right.$
3. $A \rightarrow B : \left\{ \{A, N_B\}_{Kab} \right.$
4. $B \rightarrow A : \left\{ \{B, N_A\}_{Kab} \right.$

6 Protocol: pub only

1. $A \rightarrow B : \left\{ \{A, N_A, K_{AB}\}_{pub(B)} \right.$
2. $B \rightarrow A : \left\{ \{N_A, N_B\}_{pub(A)} \right.$
3. $A \rightarrow B : \left\{ \{N_B\}_{pub(B)} \right.$
4. $B \rightarrow A : \left\{ \{h(Kab)\}_{pub(A)} \right.$

7 Protocol: Ely the frog

1. $A \rightarrow B : \{A, B, N_A\}_{pub(B)}$
2. $A \rightarrow S : \{|A, B, T_A, K_{AB}|\}_{K_{AS}}$
3. $S \rightarrow B : \{|A, B, T_S, K_{AB}|\}_{K_{BS}}$
4. $B \rightarrow A : \{|B, A, N_A + 1|\}_{K_{AB}}$

We can reduce the cost on the last message using the hash function :

1. $A \rightarrow B : \{A, B, N_A\}_{pub(B)}$
2. $A \rightarrow S : \{|A, B, T_A, K_{AB}|\}_{K_{AS}}$
3. $S \rightarrow B : \{|A, B, T_S, K_{AB}|\}_{K_{BS}}$
4. $B \rightarrow A : \{B, N_A + 1, h(K_{AB})\}_{pub(A)}$

8 Ely the little frog

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

9 Ely the big frog

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

10 Protocol Description

10.1 Ely the big frog

This protocol commences with entity A generating a nonce, denoted as N_A, which is subsequently encrypted using a perfectly random and fresh session key, K_{AB}. A then encrypts this cipher nonce, along with her identity, and transmits it to B using his public key $pub(B)$. The transmitted data is structured as follows: $\{A, \{|N_A|\}_{K_{AB}}\}_{pub(B)}$. This message costs 65.

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 to confirm the key λ and the session key K_{AB}. The transmitted data is structured as follows: $\{|B, \tau, \lambda, K_{AB}|\}_{K_{AS}}$. This message costs 166.

Then S, using the shared key K_{BS}, sends to B essentially the same message, but with A replaced by B. The transmitted data is structured as follows: $\{|A, \tau, \lambda, K_{AB}|\}_{K_{BS}}$. This message costs: 166

B receives the message $\{|A, \tau, \lambda, K_{AB}|\}_{K_{BS}}$ 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.

Then B respond to the first message of A, he can decrypt the nonce $\{|N_A|\}_{K_{AB}}$ 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.

The total cost is: **409**.