Secure Chat application for personal communication

Carignani G., Rossi F.

June 11, 2018

1 Protocol

1.1 Enrollment procedure

If an user wants to join the system it must be certified by the TTP by sending a certificate signing request (CSR) to it, containing its personal information and contacts (in order to let the TTP correctly identifying the issuer). The overall enrollment process follows this scheme:

1. An user generates a pair of private and public keys s_U, p_U
2. Then it generates a *certificate signing request* (CSR) containing identity information; in particular this request will contain its nickname in the `CN` field, that will be used by other users to initiate chat session with him. This request will be sent to a *trusted third party.*
3. Upon receiving the request, the TTP will firstly verify whether the CSR identity is confirmed and then will proceed to sign the CSR, sending back a certificate to the user. In particular, the TTP will check if the `CN` field is not already taken, to guarantee the uniqueness of the nickname system-wide.
4. At this point, with the signed certificate, the user has joined the system, identified by its nickname (or `CN`).

See Figure 3 for the sequence diagram.

1.2 Key establishment

Suppose B(ob) wants to communicate with A(lice). As described in Figure 1, he can ask her to send him her certificate, obtained as in Sub-section 1.1. Then he generates a random key k_{AB} that will be used for symmetric encryption; he builds the following request message and sends it to A:

$$m = L_{BA}||t_B||S(s_B, L_{BA}||t_B) \tag{1}$$

Where:

- E is a public key encryption operation
- p_A is A public key
- $L_{BA} =< \text{"}Bob\text{"}, \text{"}Alice\text{"}, C_B, E(p_A, k_{AB}) >$
- t_B is the timestamp associated to that message
- S is a digital signature operation
- s_B is B private key

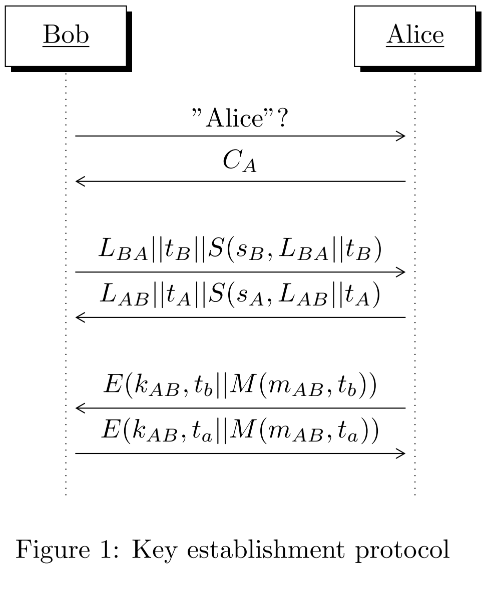
Bob
Alice
"Alice"?
C_A
$L_{BA}||t_B||S(s_B, L_{BA}||t_B)$
$L_{AB}||t_A||S(s_A, L_{AB}||t_A)$
$E(k_{AB}, t_b||M(m_{AB}, t_b))$
$E(k_{AB}, t_a||M(m_{AB}, t_a))$

Figure 1: Key establishment protocol

Upon receiving that, A builds its own request message, generating a random key m_{AB} that will be used in MACs, and sends it to B:

$$m = L_{AB}||t_A||S(s_A, L_{AB}||t_A) \tag{2}$$

Where (same fields as before are not reported):

- p_B is B public key
- $L_{AB} =< "Alice", "Bob", C_A, E(p_B, m_{AB}) >$
- t_A is the timestamp associated to that message
- s_A is A private key

Now, the keys k_{AB} and m_{AB} have been shared by the two parties. Then, B sends back to A its challenge, encrypting it by means of shared key and adding the message authentication code:

$$m_{cb} = E(k_{AB}, t_A||M(m_{AB}, t_A)) \tag{3}$$

Where E is a symmetric encryption operation, M is a MAC operation and t_A is the timestamp B has received before. Same operation is done by A:

$$m_{ca} = E(k_{AB}, t_B||M(m_{AB}, t_B)) \tag{4}$$

At this point, A and B check whether the received challenges are equal to the timestamps saved at the beginning, to be guaranteed of both freshness and correctness of the algorithm.

1.3 Message exchanging

Once a session key has been established between the two parties as in Sub-section 1.2 each message exchanged by the two parties will be built as follow:

$$A \to B : E(k_{AB}, m||t||M(m_{AB}, m||t)) \tag{5}$$

$$B \to A : E(k_{AB}, m||t||M(m_{AB}, m||t)) \tag{6}$$

Where t is a timestamp used to avoid replay attacks. Obviously we must consider a certain tolerance that takes into account the latency experienced in the network and clock misalignment.

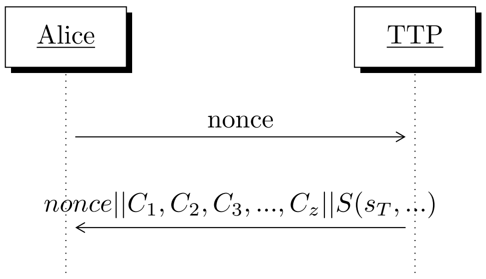

Figure 2: CRL request

1.4 Certificate revocation list (CRL)

Periodically the TTP will be able to provide a certification revocation list to users, that will be used to reject requests from potentially compromised users. The CRL will contain a list of certificate that corresponds to potentially compromised private keys. Upon receiving a request, if the contained certificate belongs to the CRL set, the request will be discarded.

The procedure followed for this step is shown in Figure 2

2 Security proofs

The most critical aspect of security in this protocol is the *key reinstallation* attack; in this attack an adversary able to eavesdrop the session key k_{AB}, attempts to impersonate one of the two parts, by means of *protocol spoofing*, and induce the other party to use the eavesdropped key.

This can be achieved by means of two possible attacks:

- *record and replay* the whole key-exchange phase
- *request forging* with the eavesdropped key

2.1 Record and replay attack

Suppose that T(rudy) has managed in some way to obtain the key k_{AB} and tries to replay the left part (referring to 1) of the key exchange protocol. Upon receiving the replayed request, A(lice) will check the timestamp to verify the freshness of the request.

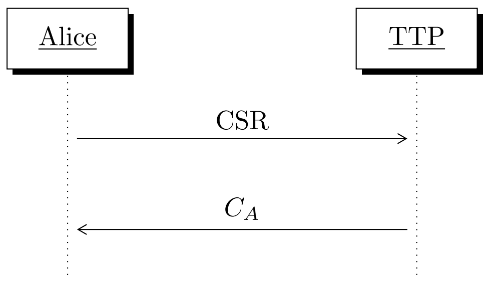

Figure 3: Certificate signing

Since T has simply replayed previous request, the timestamp check will fail and Alice will reject the request. Note that T may change the timestamp to make the request fresh; this type of attack falls back into the following category.

2.2 Request forging attack

Now, suppose that T(rudy) intercepts a legit request, changes

$$L_{BA} =< B, A, C_B, E(p_A, k_{AB}^{new}) >$$

with

$$L'_{BA} =< B, A, C_B, E(p_A, k_{AB}^{old}) >$$

and then forwards the request to A(lice).

The timestamp check will surely pass, since the request is actually fresh, but when A checks for the request signature to match the computed one, she will notice the corruption, thus rejecting the request.

Note that this property holds if.f the long term secret of B(ob) is not compromised, otherwise T could recompute the correct signature and substitute it into the request.

3 BAN logic validation

3.1 Hypotheses

3.1.1 Key block

$$A \mid\equiv \overset{k_t}{\longmapsto} T \tag{H.1}$$

$$B \mid\equiv \overset{k_t}{\longmapsto} T \tag{H.2}$$

$$B \mid\equiv A \overset{k_{ab}}{\longleftrightarrow} B \tag{H.3}$$

$$A \mid\equiv A \overset{m_{ab}}{\longleftrightarrow} B \tag{H.4}$$

3.1.2 Freshness block

For this part, we must assume that the clock synchronization is part of the *trusted computing base* of the systems upon which the protocol will run.

$$A, B \mid\equiv \#(t_b) \tag{H.5}$$

$$A, B \mid\equiv \#(t_a) \tag{H.6}$$

$$A \mid\equiv \#(C_b) \tag{H.7}$$

$$B \mid\equiv \#(C_b) \tag{H.8}$$

$$A, B \mid\equiv \#(C_t) \tag{H.9}$$

NOTE: Assuming that certificates are always fresh is only a BAN-related assumption to support validation. In practice, during implementation we <u>MUST</u>:

- Check whether certificate is expired or not (for C_a, C_b, C_t)
- Check whether certificate belongs to a *CRL* (for C_a, C_b)

3.1.3 Trust block

$$A \mid\equiv T \Rightarrow \overset{k_b}{\longmapsto} B \tag{H.10}$$

$$B \mid\equiv T \Rightarrow \overset{k_a}{\longmapsto} A \tag{H.11}$$

$$A \mid\equiv B \Rightarrow A \overset{k_{ab}}{\longleftrightarrow} B \tag{H.12}$$

$$B \mid\equiv A \Rightarrow A \overset{m_{ab}}{\longleftrightarrow} B \tag{H.13}$$

3.2 Objectives

$$A \mid\equiv \#(A \overset{k_{ab}}{\longleftrightarrow} B) \tag{O.1}$$

$$A \mid\equiv A \overset{k_{ab}}{\longleftrightarrow} B \tag{O.2}$$

$$B \mid\equiv \#(A \overset{m_{ab}}{\longleftrightarrow} B) \tag{O.3}$$

$$B \mid\equiv A \overset{m_{ab}}{\longleftrightarrow} B \tag{O.4}$$

$$B \mid\equiv A \mid\equiv A \overset{k_{ab}}{\longleftrightarrow} B \tag{O.5}$$

$$A \mid\equiv B \mid\equiv A \overset{m_{ab}}{\longleftrightarrow} B \tag{O.6}$$

3.3 Idealized protocol

M1 The first message is ignored because it is not encrypted

M2 $A \rightarrow B : C_a = \left\{A, \overset{k_a}{\longmapsto} A, t, \left\{A, \overset{k_a}{\longmapsto} A, t\right\}_{k_t^{-1}}\right\}$

M3 $B \rightarrow A : \left\{C_b, t_b, \left\{A \overset{k_{ab}}{\longleftrightarrow} B\right\}_{k_a}, \{\circ\}_{k_b}\right\}$

M4 $A \rightarrow B : \left\{C_a, t_a, \left\{A \overset{m_{ab}}{\longleftrightarrow} B\right\}_{k_b}, \{\circ\}_{k_a}\right\}$

M5 $B \rightarrow A : \left\{t_b, A \overset{k_{ab}}{\longleftrightarrow} B, A \overset{m_{ab}}{\longleftrightarrow} B, \{t_b\}_{m_{ab}}\right\}$

M6 $A \rightarrow B : \left\{t_a, A \overset{k_{ab}}{\longleftrightarrow} B, A \overset{m_{ab}}{\longleftrightarrow} B, \{t_a\}_{m_{ab}}\right\}$

3.4 Validation

M1 ......

M2 By using the message meaning rule we can say that:

$$B \mid\equiv T \mid\sim \overset{k_a}{\longmapsto} A$$

Certificate freshness holds under hypothesis H.9

$$B \mid\equiv T \mid\equiv \overset{k_a}{\longmapsto} A$$

By applying the Jurisdiction rule joined with H.1 we obtain:

$$B \mid\equiv \overset{k_a}{\longmapsto} A$$

M3 Thanks to the timestamp t_b we can say that:

$$A \mid\equiv \#(C_b, t_b, A \overset{k_{ab}}{\longleftrightarrow} B, \{\circ\}_{k_b^{-}1})$$

And, in particular (**key freshness**):

$$A \mid\equiv \#(A \overset{k_{ab}}{\longleftrightarrow} B)$$

Fulfilled O.1
Also, by applying same considerations on certificate as in previous message, we obtain:

$$A \mid\equiv \overset{k_b}{\longmapsto} B$$

By using the message meaning rule we can also say that:

$$A \mid\equiv B \mid\sim (A \overset{k_{ab}}{\longleftrightarrow} B)$$

Thanks to the freshness brought by t_b:

$$A \mid\equiv B \mid\equiv (A \overset{k_{ab}}{\longleftrightarrow} B)$$

Applying the Jurisdiction rule with H.12:

$$A \mid\equiv A \overset{k_{ab}}{\longleftrightarrow} B$$

Fulfilled O.2

M4 Thanks to the timestamp t_a we can say that:

$$B \mid\equiv \#(C_a, t_a, A \overset{m_{ab}}{\longleftrightarrow} B, \{\circ\}_{k_b^{-}1})$$

And, in particular (**key freshness**):

$$B \mid\equiv \#(A \overset{m_{ab}}{\longleftrightarrow} B)$$

Fulfilled O.3
Also, by applying same considerations on certificate as in previous message, we obtain:

$$B \mid\equiv \overset{k_a}{\longmapsto} A$$

By using the message meaning rule we can also say that:

$$B \mid\equiv A \mid\sim (A \overset{m_{ab}}{\longleftrightarrow} B)$$

Thanks to the freshness brought by t_a:

$$B \mid\equiv A \mid\equiv (A \xleftrightarrow{m_{ab}} B)$$

Applying the Jurisdiction rule with H.13:

$$B \mid\equiv A \xleftrightarrow{m_{ab}} B$$

Fulfilled O.4

M5

$$B \mid\equiv A \mid\sim (A \xleftrightarrow{k_{ab}} B)$$

$$B \mid\equiv A \mid\sim (A \xleftrightarrow{m_{ab}} B)$$

Thanks to the freshness stated by H.5:

$$B \mid\equiv \#(A \xleftrightarrow{k_{ab}} B)$$

$$B \mid\equiv \#(A \xleftrightarrow{m_{ab}} B)$$

By applying the nonce verification rule:

$$B \mid\equiv A \mid\equiv (A \xleftrightarrow{k_{ab}} B)$$

Fulfilled O.5

M6

$$A \mid\equiv B \mid\sim (A \xleftrightarrow{k_{ab}} B)$$

$$A \mid\equiv B \mid\sim (A \xleftrightarrow{m_{ab}} B)$$

Thanks to the freshness stated by H.6:

$$A \mid\equiv \#(A \xleftrightarrow{k_{ab}} B)$$

$$A \mid\equiv \#(A \xleftrightarrow{m_{ab}} B)$$

By applying the nonce verification rule:

$$A \mid\equiv B \mid\equiv (A \xleftrightarrow{m_{ab}} B)$$

Fulfilled O.6