12.2 Challenge – Response Identification (C-R-Identif.)

Basic ideas:

1. A (the claimant) proves her identity to B (the verifier) by demonstrating knowledge of a secret known only to her without revealing the secret.
2. The response is requested by a time-varying challenge.
3. The response from one execution does not provide information for a subsequent identification, as subsequent challenges will differ.

12.2.1 C-R-Identif. by symmetric key encryption

Techniques from ISO/IEC 9798-2 are described

Notation:
- E_K : symmetric encryption alg with key K
- t_A : time stamp generated by A
- r_A, r_B : random numbers by A, B
- $A \to B$: A transmits s.th. to B
- (,) : concatenation
- $\cdot^*$: optional elements are/may be added to „·"

• Unilateral authentication, time-stamp-based

$$A \to B : E_K(t_A, B^*)$$

Including the identifier B ~~means~~ prevents an adversary from re-using the message immediately on A

• Unilateral authentication, random numbers

$$A \leftarrow B : r_B \quad (1)$$

$$A \to B : E_K(r_B, B^*) \quad (2)$$

B decrypts (2), verifies r_B from (1). Inclusion of B avoids a reflection attack.

$O \leftarrow B : r_B$ (first protocol)

$O \rightarrow B : r_B$ (opening a 2nd protocol)

$O \leftarrow B : E_K(r_B, A^*)$ (in the 2nd protocol)

$O \rightarrow B : E_K(r_B, B^*)$ (in the first protocol)

Avoided by including A^* or B^*

Use K_1, K_2 or (as in the protocol) include names (A^* and B^*) and don't use same names.

A is not involved at all.

- Mutual authentication, random numbers

$A \leftarrow B : r_B$ (1)

$A \rightarrow B : E_K(r_A, r_B, B^*)$ (2)

$A \leftarrow B \quad E_K(r_B, r_A)$ (3)

B decrypts (2), verifies r_B from (1), obtains r_A, encrypts $(r_B, \ldots$

A decrypts (3), verifies r_B and r_A.

r_A might be used as shared secret key

12.2.2 C-R-Identif. by public-key techniques

Principle: The claimant decrypts a challenge encrypted by his public key.

Notation: h: Hash fct, E_A encryption under A's public key

$A \leftarrow B : h(r_B), B, E_A(r_B, B)$ (1)

$A \rightarrow B : r_B$

B chooses a random r_B, computes the witness $h(r_B)$ without revealing r_B, computes the challenge $E_A(r_B, B)$.

A decrypts $E_A(r_B, B)$ to recover r', B', computes $h(r')$.

If $h(r') = h(r_B)$ and $B' = B$ then A sends $r' = r_B$ to B.

12.2.3 C-R. Identif. based on digital signatures

Principle: The claimant signs a challenge digitally.

Notation: S_A : signature by A

$cert_A$: certificate which contains the authentic public signature key.

Protocols are from ISO/IEC 9798-3

- Unilateral with timestamps

$A \to B$: $cert_A, t_A, B, S_A(t_A, B)$

B verify that the timestamp is acceptable, the correct identifier B checks that the signature over (t_A, B) is correct.

- Unilateral with random numbers

$A \leftarrow B$: r_B

$A \to B$: $cert_A, r_A, B, S_A(r_A, r_B, B)$

B verifies its own identifier, checks validity of A's signature over (r_A, r_B, B)

- Mutual authentication with random numbers

$A \leftarrow B$: r_B

$A \to B$: $cert_A, r_A, B, S_A(r_A, r_B, B)$

$A \leftarrow B$: $cert_B, A, S_B(r_B, r_A, A)$

B verifies as above. A knows r_A, r_B verifies the validity of B's signature over (r_B, r_A, A)

12.3 Kerberos

Kerberos : three headed dog guarding the entrance to the underworld in Greek mythodology.

Grew out of a larger „Athena" at MIT

Purpose: to provide strong levels of authentication and security in key exchange between servers and clients in a network

Use symmetric encryption and relies on a trusted authority (TA)

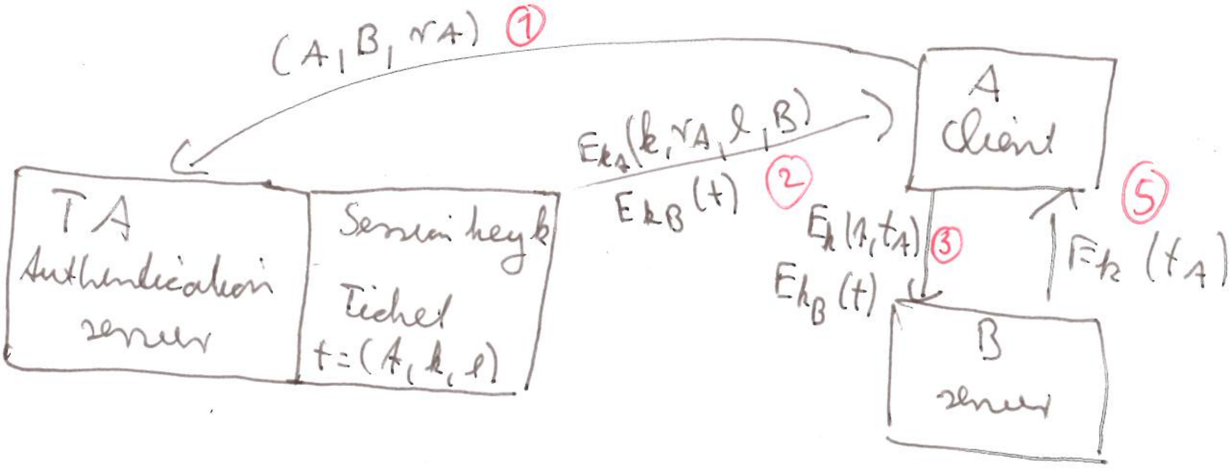
TA : central server as trusted authority, Kerberos authentication server. It knows the secret key of each client and server.

<u>Notation</u>: E_k : encryption with key k

r_A : random number by A

t_A : timestamp by A

Client A requests access to a server B. Basic actions:



Protocol actions (simplified)

1. $A \to TA$: (A, B, r_A)
2. TA generates session key k, validity period l, Ticket $t = (A, k, l)$
   $TA \to A : E_{k_A}(k, r_A, l, B), E_{k_B}(t)$
3. A recovers k, r_A, l, B, verifies r_A, B, with t_A : current time
   $A \to B : E_{k_B}(t), E_k(A, t_A)$
4. B recovers $t = (A, k, l)$ A, t_A and checks
   a) A from t matches A
   b) t_A is fresh
   c) t_A is in the validity period l

If all checks pass, A's authentication is accepted.
Additionally, to authenticate B to A

5. $B \to A$: $E_k(t_A)$
6. A recovers t_A, checks if t_A is correct. If yes, B is authenticated

Session key k is used for encrypting comm. between A and B

Remarks:

- r_A in 1 allows authentication of TA to A. (Is TA active and alive?
- t_A in 4 prevents replay attack of $E_k(A), E_{k_B}(t)$
- Secure and synchronized clocks are needed
- The full version of Kerberos includes another server, the ticket granting server.

12.4 Zero-Knowledge Identification Protocols

Disadvantage of

- fixed passwords: upon intercepting the password the owner can be impersonated.

  Ex.: Faked ATM (automatic teller machine). Bank card inserted, PIN typed in, ATM answers "card not accepted"

  But: counterfeit bank card was made, PIN was intercepted, Money was withdrawn from a legitimate ATM.

- C-R-protocols: time variant identification. Partial information shall be revealed

Zero-Knowledge protocols:

Prover A demonstrates knowledge of a secret to verifier B while revealing no information whatsoever