12.2 Challenge-Response Identification (C-R-Ident.)

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-Ident. 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 $A$, $B$
- $A \rightarrow B$: $A$ transmits n.th. to $B$
- $(\cdot)$: concatenation
- $\ast$: optional elements are/should be added to $\cdot$

- Unilateral authentication, time-stamp-based
  
  $A \rightarrow B : E_K (t_A, B^\ast)$
  
  Including the identifier $B$ means prevents an adversary from re-using the message immediately on $A$.

- Unilateral authentication, random numbers

  $A \leftarrow B : r_B \quad (1)$
  
  $A \rightarrow B : E_K (r_B, B^\ast) \quad (2)$

  $B$ decrypts (2), verifies from (1). Inclusion of $B$ avoids a reflection attack.
0 \rightarrow B : r_B \quad \text{(kid protocol)}
0 \rightarrow B : r_B \quad \text{(opening a 2nd protocol)}
0 \leftarrow B : E_k(r_B, A^*) \quad \text{(in the 2nd protocol)}
0 \rightarrow B : E_k(r_B, B^*) \quad \text{(in the 1st protocol)}

Avoided by including \( A^* \) or \( B^* \)
the \( K_1, K_2 \) 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 number:

\[ A \leftarrow B : r_B \quad (1) \]
\[ A \rightarrow B : E_k(r_A, r_B, B^*) \quad (2) \]
\[ A \leftarrow B : E_k(r_B, r_A) \quad (3) \]

\( B \) decrypts \( (2) \) (verifies \( r_B \) from \( (1) \), obtains \( r_A \), encrypts \( B^* \))
\( A \) decrypts \( (3) \) (verifies \( r_B \) and \( r_A \)).
\( r_A \) might be used as shared secret key.

12.2.2 C-R - Identity by public-key techniques

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

Notation:
\( h \): Hash fun, \( E_A \) encryption under \( A \)'s public key

\[ A \leftarrow B : h(r_B), B, E_A(r_B, B) \quad (1) \]
\[ A \rightarrow B : r_B \]

\( B \) chooses a random \( r_B \), computes the within \( h(r_B) \) without revealing \( r_B \), computes the challenge \( E_A(r_B, B) \).
\( A \) decrypts \( E_A(r_B, B) \) to recover \( r' \), \( B' \), compute \( h(r') \).
If \( h(r') = h(r_B) \) and \( B' = B \) then \( A \) sends \( r' = r_B \) to \( B \).
12.2.3 C-R - Isolated based on digital signatures

**Principle:** The claimant signs a challenge digitally.

**Notation:**
- \( S_A \): signature by \( A \)
- \( \text{cert}_A \): certificate which contains the authentic public signature key.

Protocols are from ISO/IEC 9798-3

6. Unilateral with timestamps

\[ A \rightarrow B : \text{cert}_A, t_A, B, S_A(t_A, B) \]
- B verifies that the timestamp is acceptable, the correct identifier B checks that the signature over \((t_A, B)\) is correct.

7. Unilateral with random number

\[ A \rightarrow B : \; N_B \]
\[ A \rightarrow B : \text{cert}_A, N_A, B, S_A(N_A, N_B, B) \]
- B verifies its own identifier, checks validity of A's signature over \((N_A, N_B, B)\).

8. Mutual authentication with random number

\[ A \rightarrow B : \; N_B \]
\[ A \rightarrow B : \text{cert}_A, N_A, B, S_A(N_A, N_B, B) \]
\[ A \rightarrow B : \text{cert}_B, A, S_B(N_B, N_A, A) \]
- B verifies as above. A knows \( N_A, N_B \) verifies the validity of B's signature over \((N_B, N_A, A)\).
Kerberos: three headed dog guarding the entrance to the underworld in Greek mythology.

Created 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

The symmetric encryption and relies on a trusted authority (TA).

TA: central server as trusted authority, Kerberos authentication

Notation:

- $k$: encryption with key $k$
- $R_A$: random number by $A$
- $T_A$: timestamp by $A$

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

- $A \rightarrow TA$: request for a ticket
- $TA \rightarrow A$: ticket and ciphertext $C$ with $k_{TA}$
- $A \rightarrow B$: ciphertext $C$ and ticket
- $B \rightarrow A$: successful authentication with ticket and $C$
Protocol: Kerberos

1. $A \rightarrow TA: (A, B, r_A)$
2. $TA$ generates session key $k_S$, validity period $l$, ticket $t = (A, k_S, l)$
   $TA \rightarrow A: E_k_S (k_S r_A, l, B), E_k_B (t)$
3. $A$ recovers $k_S, r_A, l, B$, verifies $r_A, B$, with $t_A$ to cancel this
   $A \rightarrow B: E_k_B (t), E_k_S (A, t_A)$
4. $B$ recovers $t = (A, k_S, l)$ $A, t_A$ and checks
   a) $A$ from $t$ matches $A$
   b) $t_A$ is fresh
   c) $t$ is in the validity period $l$
   If all checks pass, $A$'s authentication is accepted.
   Additionally, to authenticate $B$ to $A$
5. $B \rightarrow A: E_k (t_A)$
6. $A$ recovers $t_A$, checks if $t_A$ is correct. If yes, $B$ is authenticated
   Session key $k_S$ is used for encrypted comm. between $A$ and $B$

Remarks:
- $r_A$ in $t$ allows authentication of TA to $A$. Is TA address available?
- $t_A$ in $t$ prevents replay attacks of $E_k (A), E_k_B (t)$
- Secure and synchronized clocks are needed.
- The full $k$ version of Kerberos includes another server, the ticket granting server.
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: true 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.