12.4 Zero-Knowledge Identification Protocols

Disadvantages of

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

Ex. : Faked ATM : Bank card inserted, PIN typed in, ATM answers „card not accepted"

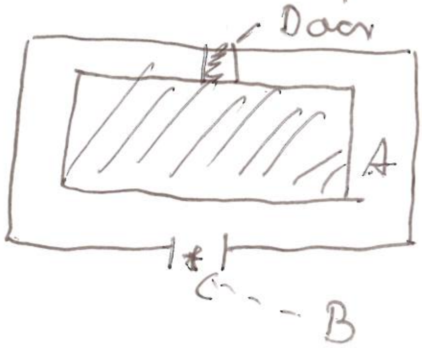
But: Counterfeit bank card was made, PIN was intercepted. Money was withdrawn from a legitimate ATM

- 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

Demonstrative Example



A proves to B that she can unlock the door (without giving away any information how she does it)

- A enters the tunnel and goes to the left or to the right
- B waits, stands at *, and calls randomly „left" or „right"
- A appears from the left or the right, as requested
- If A comes from the right direction for each of n repetitions there is only a probability of 2^{-n} that she does not know how to open the door.
- O/E sets up a video camera at *, will gain no information to convince others that O/E can go through the door.

General structure of zero-knowledge protocols

1. $A \to B$: witness: A selects a random element, from this computes a public witness: Purpose
   - variation from other protocol runs
   - defines a set of questions, answerable only by A
2. $A \leftarrow B$: challenge: B selects a question
3. $A \to B$: response: A answers the question, B checks correctness

Example: Let $n = p \cdot q$ $\quad p \neq q$ prime

A selects random s, computes $y = s^2 \bmod n$ with $\gcd(y, n) = 1$

A claims to know a square root of y without revealing s.

Protocol:

1. A chooses randomly r_1, r_2 with
   $$r_1 \cdot r_2 \equiv s \pmod n$$
   Choose r_1 at random with $\gcd(r_1, n) = 1$ and calculate $r_2 = r_1^{-1} \cdot s \bmod n$
   Compute $x_1 = r_1^2 \bmod n$ $\quad x_2 = r_2^2 \bmod n$
   $A \to B$: (x_1, x_2) (witness)
2. B checks, if $x_1 \cdot x_2 \equiv y \pmod n$
   B chooses x_1 or x_2 randomly
   B asks A to supply a square root of it. (Challenge)
3. A sends the square root, e.g., r_1 to B
   B checks if it is a square root by $r_1^2 \equiv x_1 \pmod n$

Iterate this protocol t times, because O/E have a 50% chance of giving a correct answer.

Ex.: Discuss the protocol.

12.4.1 / Feige – Fiat – Shamir Identification Protocol (1988)

Relies on the hardness of computing square roots modulo n, n composite

Objective : A proves her identity to B

System parameters

(i) A, TA (Trusted Authority), publishes $n = p \cdot q$ $p \neq q \equiv 3 \pmod 4$

(ii) Each entity A selects random numbers $s_1, \dots, s_k \in \{1, \dots, n-1\}$ $\gcd(s_i, n) = 1$, computes $v_i = (s_i^2)^{-1} \bmod n$ publishes $v_1, \dots, v_k$

Protocol actions

1. A chooses a random integer r, compute $x = r^2 \bmod n$

   $A \to B : x$ (witness)

2. B chooses random bits $b_1, \dots, b_k \in \{0, 1\}$

   $A \leftarrow B : (b_1, \dots, b_k)$ (challenge)

3. A computes : $y = \left(r \prod_{j=1}^{k} s_j^{b_j} \right) \bmod n$

   $A \to B : y$ (response)

4. B checks that $y^2 \prod_{j=1}^{k} (v_j)^{b_j} \equiv x \pmod n$

Security aspects

Oscar wants to impersonate A.

Suppose O guess $(b_1, \dots, b_k)$ before he sends x:

O chooses a random integer $a \in \{1, \dots, n-1\}$ computes

$x = a^2 \prod_{j=1}^{k} v_j^{b_j} \bmod n$

O sends in step 3 $O \to B : a$

B checks in 4 that $a^2 \prod_{j=1}^{k} v_j^{b_j} \equiv x \pmod n$ accepts A's identity

However the probability to guess $(b_1, \dots, b_k)$ correctly in t trials is $\frac{1}{2^{tk}}$

An identification scheme based on the FFS identification protocol:

I_A : identification string for A, containing, e.g., name, birthday, etc.

Notation: $I_A \| j$ concatenation ; h some hash function

TA computes $h(I_A \| j)$ for some j until it receives integers which are square roots

$v_1 = h(I_A \| j_1), \ldots, v_k = h(I_A \| j_k)$ and

$s_1, \ldots, s_k$ are computed by knowing p, q.

$I_A, n, j_1, \ldots, j_k$

$s_1, \ldots, s_k$ are given to A and kept secret

Identification to an ATM, e.g.,

- ATM reads I_A from A's card
- download $n, j_1, \ldots, j_k$ from a data base
- calculate $v_1 = h(I_A \| j_1), \ldots, v_k = h(I_A \| j_k)$
- perform the preceding protocol t times

12.4.2) Schnorr Identification Protocol

Obj.: A proves her identity to B

Relies on hardness of computing discrete logs.

System parameters

1. A trusted authority chooses:
   - p prime, q prime, $q \mid p-1$ $\quad (p \approx 2^{1024}, q \geq 2^{160})$
   - $\beta \in \mathbb{Z}_p^*$ of order q
   - TA publishes and signs (p, q, β)
   - Security parameter t with $2^t < q$ $\quad$ e.g., $t \geq 40$
2. Each user A
   - chooses a private key a $\quad 0 \leq a \leq q-1$
   - computes $v = \beta^{-a} \mod p$
   - publishes v $\quad$ (TA signs (A, v) after securing the identity of A)

Protocol actions

1. A chooses a random number $r \in \{1, \dots, q-1\}$

   $A \to B: \; X = \beta^r \mod p$ $\quad$ (witness)
2. B chooses a random number $e \in \{1, \dots, 2^t\}$

   $A \leftarrow B: \; e$ $\quad$ (challenge)
3. A checks that $1 \leq e \leq 2^t$

   $A \to B: \; Y = (a \cdot e + r) \mod q$ $\quad$ (response)
4. B computes $z = \beta^Y \cdot v^e \mod p$

   verifies $z = X$ $\quad$ (the identity of A)

Remarks

a) Protocol is correct since

$$\beta^{y} \cdot v^{e} \equiv \beta^{(a \cdot e + r) \bmod q} \, \beta^{-a \cdot e} \overset{(*)}{\equiv} \beta^{r} \equiv x \pmod{p}$$

(*) this is true as β has order q in $\mathbb{Z}_p^*$, cf. DSA

b) Suppose O/E guesses e prior to sending x

O chooses some y, compute $x = \beta^{y} \cdot v^{e} \bmod p$, sends

in 1: $O \to B$: x

in 3: $O \to B$: y

Then $z \equiv \beta^{y} \cdot v^{e} \equiv x \pmod{p}$, B accepts in 4 O or A's identity

c) The protocol is particularly suited for smart cards

Computational effort:

in 1: fast exponentiation (expensive, but may be computed in advance)

in 3: one modular multiplication and addition (cheap!)