Tutorial IX group 2

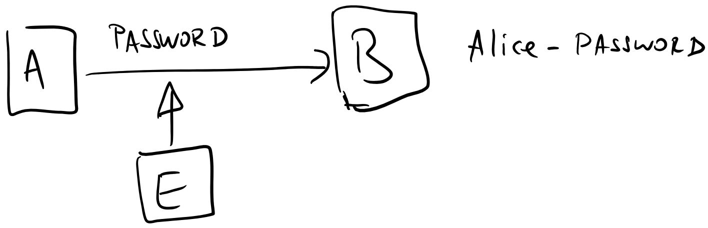
14 November 2019 16:02

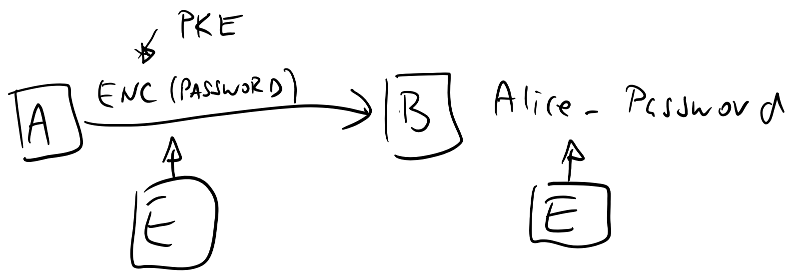
Identification

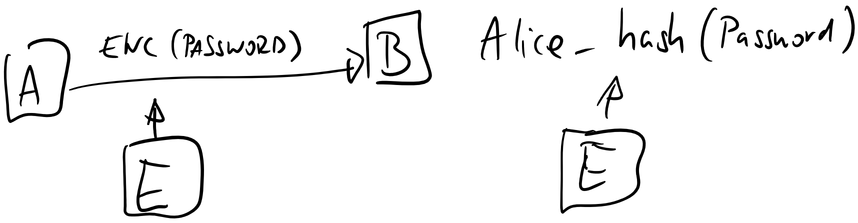
Secret sharing

message authentication → Orthogonal arrays

Identification







These work only for trusted $\boxed{B}$ (he knows the password)

Here ↓ we learn about zero-knowledge identification protocols

Alice can prove knowledge of her password to Bob without revealing it.

Alice - Prover

Bob - Verifier

Eve - Evesdropper

1.) Commitment $A \to B$

2.) Challenge $B \to A$

3.) Response $A \to B$

4.) Verification

Fiat-Shamir identification

↳ based on hardness of calculating square roots i.e $\sqrt{C}$ mod n, where $n = p \cdot q$, without knowledge of p and q.

PRIVATE: $s \in \{1, \dots, n-1\}$, $(p, q$, with $n = p \cdot q)$

PUBLIC: n, $V = s^2 \bmod n$

1.) commitment: Alice chooses random $1 \leq r < n$ and sends $x = r^2 \bmod n$ to Bob

2.) challenge: Bob chooses a random bit b and sends it to Alice

3.) response: Alice sends $y = r \cdot s^b \bmod n$ to Bob

4.) verification: Bob verifies whether $y^2 = x \cdot v^b \bmod n$

→ r needs to be random and unknown to Bob. Why?
if Bob knew r, in step 2.) he can choose $b=1$, then

$y = r \cdot s \bmod n$ and he can calculate $s = y \cdot r^{-1} \bmod n$

→ b needs to be random and unknown to Alice before her commitment. Why?

- If $\boxed{\text{a Prover}}$ knows $b=0$ can she pass the protocol?

perhaps they don't know s

verification will be $y^2 = x \bmod n$

Can you find two such numbers?

1.) choose y 2.) calculate x (order is important - in the protocol x is sent first and y second)

- —— " —— $b=1$ —— " ——

verification will be $y^2 = x \cdot v$ n

Can you find such x and y?

$x = y^2 \cdot v^{-1} \bmod n$

1.) choose y 2.) calculate x

TRANSCRIPT

(x, b, z) valid iff

$$\boxed{n = 15 \qquad v = 4}$$

$(x, 0, z) \rightsquigarrow (1, 0, 11)$ $\qquad 11^2 = 1 \mod 15$

$(x, 1, z) \rightsquigarrow (6, 1, 3)$ $\qquad 3^2 = x \cdot 4 \mod 15$

$4 \cdot 3^2 = x \mod 15$

$4 \cdot 9 = x \mod 15$

$36 = x \mod 15$

$6 = x \mod 15$

$(x, 0, z_0)$

$(x, 1, z_1)$

calculating these two transcripts is as hard as finding s

$z_0^2 = x \mod n$

$z_1^2 = x \cdot v \mod n$

$z_0 = \sqrt{x} \mod n$

$z_1 = \sqrt{x} \cdot s \mod n$

$z_1 = z_0 \cdot s \mod n$

$s = z_1 \cdot z_0^{-1} \mod n$

After n correct rounds Bob knows he is talking to Alice w.p. $1 - \frac{1}{2^n}$.

After n correct rounds Bob knows he is talking to Alice w.p $1-\frac{1}{2^n}$.

Shnorr identification

↳ based on discrete logarithm

Public: p - large prime

q - a prime dividing $(p-1)$ [q - is 140 bit]

$\alpha \in \mathbb{Z}_p^*$ of order q [$\alpha^q = 1 \bmod p$]

$\boxed{\text{Security parameter } t}$ s.t. $2^t < q$ (how hard it is to guess a challenge)

$v = \alpha^{-a} \bmod p$

$Sig_{TA}(\text{ALICE}, v, p, q, \alpha)$

PRIVATE: $1 \leq a \leq q-1$

1.) Commitment: Alice randomly chooses $0 \leq k \leq q-1$

and sends

2.) challenge: Bob chooses randomly $1 \leq r \leq 2^t - 1$

and sends it to Alice

3.) response: Alice sends

4.) verification: Bob checks $\gamma = \alpha^{y} \cdot v^{r} \bmod p$

$$\alpha^{k} = \alpha^{(k+ar)} \cdot \alpha^{-ar} \bmod p$$

$$\alpha^{k} = \alpha^{k} \mod p$$

→ k should be random and unknown to Bob

if Bob knows k then $a = (y-k) \cdot r^{-1} \mod p$

→ r should be random and unknown to $\boxed{\text{a Prover}}$

otherwise anyone can identify as Alice.

Two numbers γ and y for which [illegible] can be used as commitment and response

With knowledge of r, how hard is it to find such numbers?

EASY: 1.) choose y 2.) calculate γ

TRANSCRIPTS:

(γ, r, y) valid iff $\gamma = \alpha^{y} v^{r} \mod p$

$$\left.\begin{array}{l}(\gamma, r_1, y_1) \\ (\gamma, r_2, y_2)\end{array}\right\} \text{calculating is equivalent to calculation of } a$$

$$\alpha^{y_1} v^{r_1} = \alpha^{y_2} v^{r_2} \mod p$$

$$\alpha^{y_1} \cdot \alpha^{-a \cdot r_1} = \alpha^{y_2} \alpha^{-a r_2} \mod p$$

$$y_1 - a r_1 = y_2 - a r_2 \mod q$$

$$a = (y_1 - y_2) \cdot (r_1 - r_2)^{-1} \mod q$$

$$a = (y_2 - y_1) \cdot (r_2 - r_1)^{-1} \bmod q$$

Secret Sharing

U - user set $\quad U = \{1, \dots, n\}$

A - access structure $\quad A \subseteq P(U) = 2^U$

$$P(U) = \{\emptyset, \{1\}, \{2\}, \dots, \{n\}, \{1,2\}, \{1,3\}, \dots, U\}$$

$$|P(U)| = 2^{|U|}$$

$$U = \{A, B, C, D\}$$

$$A = \{\{A, B\}, \{B, C, D\}, \{A, C, D\}\}$$

$$A = \{\{A, B\}, \cancel{\{A, B, C\}}\}$$

Oversets are always included by default.

Threshold scheme (n, t)

n - number of users

t - the number of users required to recover the secret

$(4,2)$ - scheme

$$U = \{A, B, C, D\}$$

$$A = \{AB, AC, AD, BC, BD, CD\}$$

How to construct threshold schemes

1.) choose a (large) prime p

2.) to each user send $x_i \in \mathbb{Z}_p$ (typically $x_i = i$)

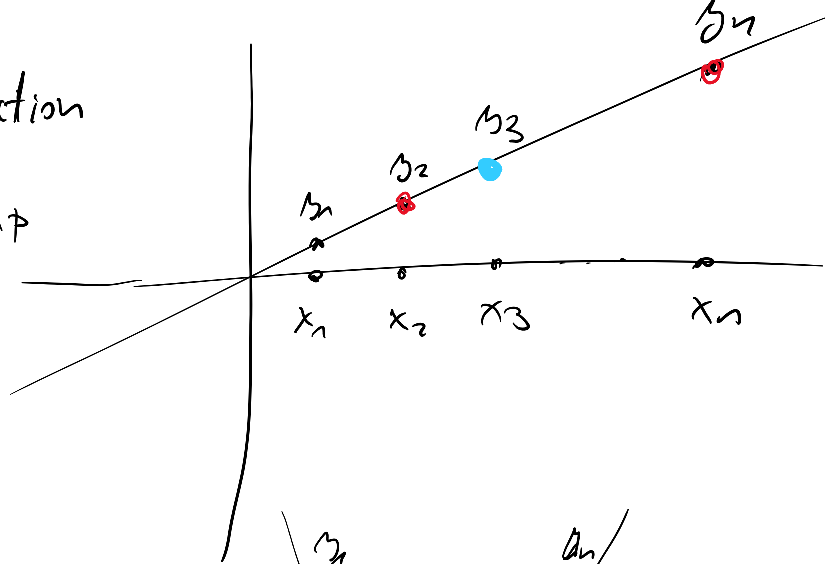
3.) to share a secret $S \in \mathbb{Z}_p$ send secretely to each user $y_i = a(x_i)$

where
$$a(x) = \left(\sum_{j=1}^{t-1} a_j x^j\right) + S \mod p$$

and a_j are chosen at random and kept secret

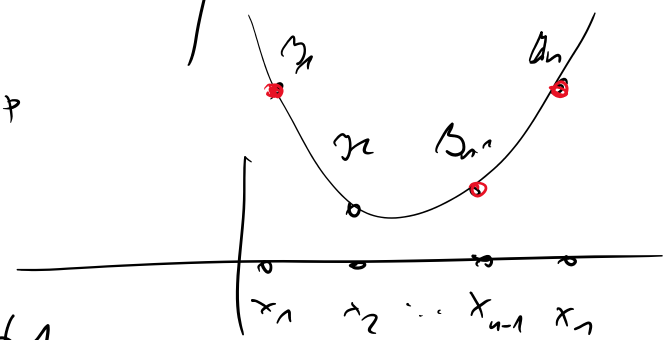
for $t=2$ a is a linear function

$$a(x) = a_1 x + S \mod p$$



for $t=3$ a is quadratic

$$a(x) = a_2 x^2 + a_1 x + S \mod p$$



for t the degree of $a(x)$ is $t-1$

and t points are required to reconstruct $a(x)$ and find S.

Example of $(3,3)$-scheme

$f(1) = 9 \mod 11$

$f(2) = 9 \mod 11$

$f(3) = 4 \mod 11$

degree of f is 2.

secret $\to b$

$f(x) = ax^2 + bx + C$

$$\boxed{\begin{aligned} a + b + c &= 9 \mod 11 \\ 4a + 2b + c &= 9 \mod 11 \end{aligned}}$$

$9a + 3b + c = 4 \mod 11$

ORTHOGONAL ARRAYS

$OA(n, k, \lambda)$ is a $\lambda n^2 \times k$ array of n symbols s.t. in any two columns of the array each of the n^2 possible pairs of symbols appear exactly λ-times.

symbols, columns, repetition

$OA(3, 3, 1)$

$\lambda n^2 \times k$

$1 \cdot 3^2 \times 3$

9×3

assume Alice sent $(m_2, 1)$

	m_1	$\boxed{m_2}$	
h_1	0	0	0
h_2	1	1	1
h_3	2	2	2
h_4	0	1	2
h_5	1	2	0
h_6	2	0	1
h_7	0	2	1

1.) Adversary wants to send message to Bob without seeing a message-tag pair sent by Alice

2.) Alice sent a valid pair $m, h(m)$

Eve wants to send

h_7	0	2	1
h_8	1	0	2
h_9	2	1	0

Eve wants to send $m', h(m')$ instead

$m \neq m'$

In authentication

$m, h(m)$

φ
hash

$m', h(m')$

$$A \xrightarrow{m,\, h_k(m)} B$$

k $\qquad$ k