Tutorial IX

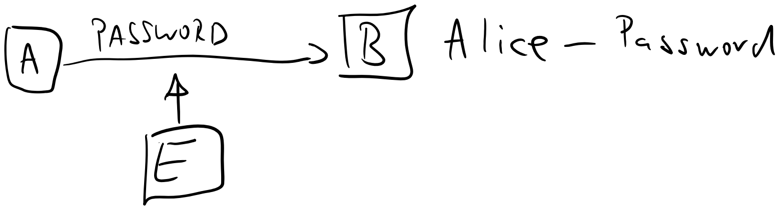
14 November 2019 10:03

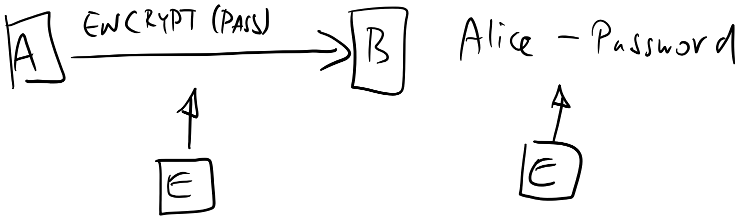
Identification

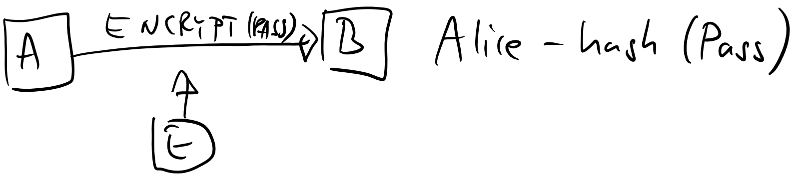
Secret Sharing

Message Authentification → Orthogonal arrays

Identification







These work only for trusted $\boxed{B}$

In principle if Alice uses the same password for different servers $\boxed{B}$ can impersonate her.

We study dynamic protocols in which $\boxed{B}$ doesn't have to be trusted. These are proofs of knowledge of passwords (zero knowledge) be trusted. These are proofs of knowledge of passwords (zero knowledge proofs)

Alice - Prover

Bob - Verifier

Eve - Eavesdropper

1.) commitment $A \to B$

2.) challenge $B \to A$

3.) response $A \to B$

4.) verification step

Fiat-Shamir identification

↳ based on hardness of calculating $\sqrt{c} \mod n$, where $n = p \cdot q$, without knowledge of p and q.

PRIVATE: $s \in \{1, \dots, n-1\}$, $(p, q$ s.t. $n = p \cdot q)$
↳ two large primes

PUBLIC: n, $v = s^2 \mod n$

1.) [redacted] Alice chooses a random $r < n$ and sends $x = r^2 \mod n$

2.) [redacted] Bob chooses a random bit b and sends

2.) [redacted] Bob chooses a random bit b and sends it to Alice

3.) [redacted] Alice replies with $\boxed{y = r \cdot s^{\boxed{b}}} \mod n$

4.) [redacted] Bob checks whether $y^2 = x \cdot v^b \mod n$

→ r needs to be random and unknown to Bob

↳ Why? Bob with knowledge of r can choose a challange $b=1$. Then $y = r \cdot s$ and $s = y \cdot r^{-1} \mod n$

→ b needs to be random and unknown to the $\boxed{\text{prover}}$.

↳ Why?

A) Assume you know $b=0$. Can you pass for Alice without knowing s?

yes. 1.) → $x = r^2 \mod n$

3.) → $y = r$

B) Assume you know $b=1$. Can you pass for Alice?

You need to find x and y s.t.

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

Can you? Choose y and calculate

$$x = y^2 \cdot v^{-1} \mod n \quad ✓$$

TRASCRIPTS

are triples:

(x, b, y)

Valid transcript: $y^2 = x \cdot v^b \mod n$

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

$(1, 1, 2) \rightsquigarrow 2^2 = 1 \cdot 4 \mod n$ ✓

$y^2 = x \cdot v^1$

$(4, 0, 2) \rightsquigarrow 4 = x \cdot 1$

$(9, 0, 3) \rightsquigarrow 9 = x \cdot 1$

$(1, 1, 2) \rightsquigarrow 4 = x \cdot 4$

$(10, 1, 5) \rightsquigarrow 5^2 = x \cdot 4 \Rightarrow 25 = x \cdot 4 \mod 15$

$4 \cdot 25 = x \mod 15$

$100 = x \mod 15$

$10 = x \mod 15$

$(x, 0, y_0)$

$(x, 1, y_1)$

$y_0^2 = x \mod n$

$y_1^2 = x \cdot v \mod n$

$y_0 = \sqrt{x} \mod n$

$y_1 = \sqrt{x} \cdot \sqrt{v} \mod n$

$y_1 = \sqrt{x}\cdot\sqrt{v} \mod n$

$y_1 = y_0 \cdot \sqrt{v} \mod n$

Each round convinces Bob he is talking to Alice w. p. of error $\frac{1}{2}$

After n rounds with correct responses Bob knows he is talking to Alice except for probability $\frac{1}{2^n}$.

Shnorr identification

↳ based on discrete logarithm

Pubic: p - large prime

q - a prime dividing $(p-1)$ $\qquad$ q - 140 bits

$\alpha \in \mathbb{Z}_p^*$ of order q

security parameter t s.t. $\boxed{2^t < q}$

Signed by an authority: $\boxed{v = \alpha^{-a} \mod p \qquad sig_{TA}(Alice, v, p, q, \alpha)}$

Private: a

1.) Alice randomly chooses $0 < k < q$ and sends $\gamma = \alpha^k \mod p$

2.) Bob chooses randomly $1 \le r \le 2^t$ and sends it to Alice

2.) [redacted] Bob chooses randomly $r = v - c$ and sends it to Alice

3.) [redacted] Alice sends $y = (k + ar) \bmod q$

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

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

$$\alpha^{k} = \alpha^{k} \bmod p \quad \checkmark$$

→ k (the commitment) needs to be random and secret (and fresh in every round)

if Bob knows k, then $a = (y - k) \cdot r^{-1} \bmod q$

→ r (the challenge) needs to be random and secret in every run

if the $\boxed{\text{Prover}}$ knows challenge beforehand (before commitment)

then she needs to calculate γ and y s.t.

$$\gamma = \alpha^{y} \cdot v^{r} \bmod p$$

which can be done by choosing [redacted] and calculating γ

Transcripts

(γ, r, y) $\qquad \gamma = \alpha^{y} \cdot v^{r} \bmod p$

(γ, r_1, y_1) $\qquad \alpha^{y_1} \cdot v^{r_1} = \gamma = \alpha^{y_2} \cdot v^{r_2} \bmod p$

(γ, r_2, y_2)

$\Downarrow$

impossible without knowing a

Secret Sharing

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

A - acces structure $\quad A \subseteq P(U) = 2^U$ (set containing all the subsets of U)

$$P(U) = \left|\{\emptyset, \{1\}, \{2\}, \ldots, \{n\}, .. \{1,2\}, \ldots \quad U\}\right| = 2^{|U|}$$

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

$A = \{\{A,B\}, \{B,C,D\}, \{A,C,D\}\}$ $\rightsquigarrow$ typically you require

$X, Y \in A$ then $X \not\subseteq Y$

$Y \not\subseteq X$

Threshold scheme (n, t)

n - number of users

t - the number of users required to reconstruct the secret

How to do this?

1.) Choose a prime p

to each user send $x_i \in \mathbb{Z}_p^*$

(typically $x_i = i$)

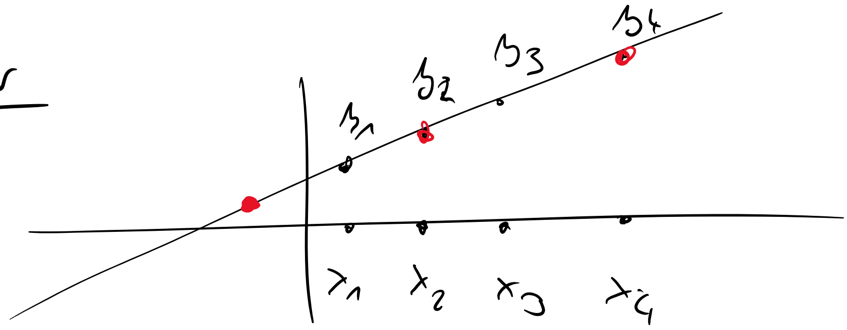
2.) To share a secret S send (secretely) to each user $y_i = a(x_i)$

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

and a_i were chosen at random and are kept secret

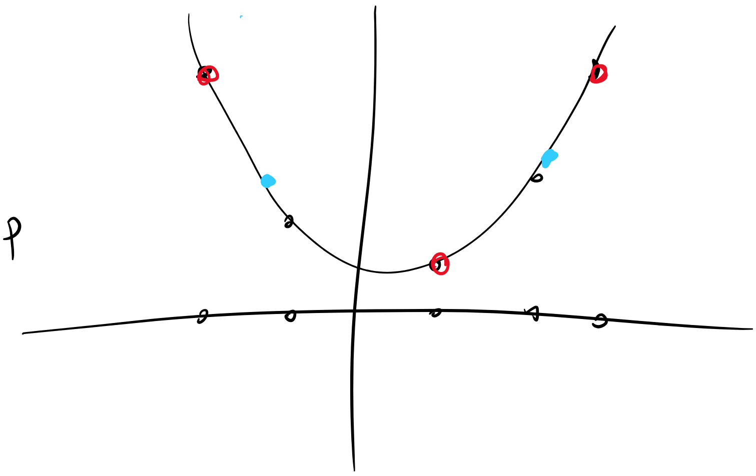
for $t=2$ a is linear

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



for $t=3$ a is quadratic

$a(X) = a_2 x^2 + a_1 x + \boxed{S} \mod p$



---

for $a(x)$ of degree $t-1$ exactly t points are needed to reconstruct the secret why?

Example:

$f(1) = 9 \mod 11$

$f(2) = 9 \mod 11$

$f(3) = 4 \mod 11$

$f(3) = 4 \mod 11$

if degree of $f(x)$ is 2, we have

$f(x) = ax^2 + bx + c \mod 11$

$f(1) = a + b + c = 9 \mod 11$

$f(2) = 4a + 2b + c = 9 \mod 11$

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

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 appears exactly λ-times

$OA(3, 3, 1)$

symbols, collumns, pair repetitions

$\lambda n^2 \times k$

$1 \cdot 9 \times 3$

	m_1	m_2	m_3
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
h_8	1	0	2

1.) Adversary wants to send a message without interception

2.) Adversary captures a valid message hash pair $(m, h(m))$ and wants to send $(m', h(m'))$ with

n_8	1	0	2
n_9	$\boxed{2}$	$\boxed{1}$	$\boxed{0}$

to send $(m', h(m'))$ with $m' \neq m$

$(m, 2)$

MESSAGE AUTHENTICATION WITH SHARED KEY

Secret shared key is used to choose a hash function h

then, $m, h_k(m)$ is sent and receiver checks

if the message he received is consistent.