Zero-Knowledge Proof Systems

Rappresentazione.

$L \subseteq \Sigma^*$, $\Sigma = \{0,1\}$ (codifica binaria dei "teoremi")

insieme di simboli

Linguaggio

$x \in L$ (teorema vero), $x \notin L$ (teorema falso)

(veridicità $\equiv$ appartenenza al linguaggio)

Ripensando alla lezione scorsa, L potrebbe essere:

$L = \{$ codifiche binarie di coppie di grafi (G_0, G_1) tali che $G_0 \simeq G_1 \}$

È un modo di:

$L \in NP$ significa che $\exists$ un alg. deterministico

$V : \Sigma^* \times \Sigma^* \to \{0,1\}$ tale che $\forall x \in L \ \exists \pi \in \Sigma^*$

tale che $V(x, \pi) = 1$ con $|\pi| < \text{poly}(|x|)$

teorema (istanza) — prova (testimone appartenenza)

prova "breve"

Perfect Zero-Knowledge

Let $L \subseteq \Sigma^*$. A zero-knowledge proof system for L is a pair (P, V) satisfying

1. (Complete) For all $x \in L$, a verifier says "yes" after interacting with the prover
2. (Sound) For all $x \notin L$, and for all provers P^*, a verifier says "no" after interacting with P^* with probability at least $1/2$.
3. (Perfect) For all verifiers V^*, there exists a simulator S^* that is a randomized polynomial time algorithm such that for all $x \in L$,

$$\{transcript((P, V^*)(x))\} = \{S^*(x)\}$$

Warm up ...

n intero positivo

a è un res. quadratico mod n se la congruenza

$y^2 \equiv a \bmod n$ ha una soluzione $y \in \mathbb{Z}_n^*$

Thm Sia $p > 2$ primo. Ogni residuo quadratico a ha esattamente 2 radici quadrate in $\mathbb{Z}_p^*$.

$\Rightarrow$ $sq : \mathbb{Z}_p^* \to \mathbb{Z}_p^*$ $\quad sq(x) = x^2 \bmod p$

è una funzione 2-a-1

insieme dei residui quadratici mod p $\longrightarrow$ $|QR_p| = \frac{|\mathbb{Z}_p^*|}{2} = \frac{p-1}{2}$

Possiamo renderci conto anche osservando che:

$\mathbb{Z}_p^*$ ciclico $\Rightarrow \exists$ generatore $g \in \mathbb{Z}_p^*$:

$$\mathbb{Z}_p^* = \{ g^0, g^1, g^2, \dots \quad g^{p-2} \}$$

elevando al quadrato $= \{ g^0, g^2, g^4, \dots g^{p-3}, g^0, g^2, g^4 \dots g^{p-3} \}$

$\Rightarrow$ ogni quadrato compare due volte.

Th (Criterio di Eulero). Sia p primo dispari. Un intero $a \in \mathbb{Z}_p^*$ è un residuo quad. mod p se e solo se

$$a^{(p-1)/2} \equiv 1 \bmod p$$

Indichiamo i non residui con QNR_p $\left(|QNR_p| = \frac{p-1}{2}\right)$

Corollario Siano $x, x' \in QR_p$ e $y, y' \in QNR_p$

$\Rightarrow$ $x \cdot x' \in QR_p$, $yy' \in QR_p$ e $xy \in QNR_p$

(tutto mod p, p primo >?)

Thm. Sia $N = p \cdot q$, p e q primi dispari distinti

$y \bmod N$ è un res. quad. **x** e solo **x** risulta

$$\begin{cases} y_p = y \bmod p \text{ un res. qu. mod } p \\ y_q = y \bmod q \text{ un res. qu. mod } q \end{cases}$$

Sia QR_N l'insieme dei res. quadratici mod N

$$|QR_N| = |QR_p| \cdot |QR_q| = \left(\frac{p-1}{2}\right)\cdot\left(\frac{q-1}{2}\right) = \frac{(p-1)(q-1)}{4}$$

Esattamente un quarto degli elementi di $\mathbb{Z}_N^*$ sono res. quadratici

Come calcoliamo le radici ?

Sia p un primo dispari. (Facile)

Possono essere due casi : $p \equiv 1 \bmod 4$, $p \equiv 3 \bmod 4$

$p \equiv 3 \bmod 4 \Rightarrow x = \pm a^{(p+1)/4} \bmod p$

$p \equiv 1 \bmod 4$ (... più complicato ma $\exists$ ppt alg per il calcolo)

Sia $N = p \cdot q$, p e q primi dispari. $(y \bmod N)$

Procediamo come segue:

- calcoliamo le due radici di $y \bmod p$, $\pm x_p$
- calcoliamo le due radici di $y \bmod q$, $\pm x_q$

le 4 radici quadrate x_1 x_2 x_3 x_4

(x_p, x_q) $(-x_p, x_q)$ $(x_p, -x_q)$ $(-x_p, -x_q)$

sono ottenibili applicando il teorema cinese del resto.

Nota: occorre conoscere la fattorizzazione di N per procedere in tal modo.

E se la fattorizzazione di N non è nota?

Posso definire formalmente il problema del calcolo e dimostrare che è tanto difficile quanto fattorizzare!

$SQR_{\mathcal{A},GenMod}(n)$

1. $\mathcal{C}$ esegue $GenMod(1^n)$ per ottenere (p, q, N)
2. $\mathcal{C}$ sceglie uniformemente a caso $y \in \mathcal{QR}_N$
3. $\mathcal{A}$ riceve da $\mathcal{C}$ la coppia (N, y) e dà a $\mathcal{C}$ il valore $x \in \mathbb{Z}_N^*$
4. Se $x^2 \equiv y \bmod N$, allora $\mathcal{C}$ dà in output 1; altrimenti, 0.

Formalmente

Definizione 8. *Relativamente a $GenMod(1^n)$, il problema del calcolo delle radici quadrate è difficile se, per ogni algoritmo ppt $\mathcal{A}$, esiste una funzione trascurabile, tale che*

$$Pr[SQR_{\mathcal{A},GenMod}(n) = 1] \leq negl(n)$$

Detto ciò, l'assunzione del calcolo delle radici quadrate può essere formulata come segue

Assunzione 2 (Calcolo delle radici quadrate)**.** *Esiste un algoritmo $GenMod(1^n)$ relativamente al quale il problema del calcolo delle radici quadrate è difficile.*

È possibile dimostrare infatti che:

Problema della Fattorizzazione e Problema del calcolo delle radici quadrate

sono equivalenti!

Non lo formalizzo, ma anche il problema

di distinguere un quadrato da un non quadrato

in $\mathbb{Z}_N^*$ è ritenuto un problema difficile

(... per i dettagli consultate gli appunti nella

parte relativa alle assunzioni crittografiche)

Sistema di prova a conoscenza zero per i residui quadratici

$N = p \cdot q$, $x \in \mathbb{Z}_N^*$. Vogliamo provare che:

p, q: primi dispari distinti di grossa taglia

$x = z^2 \bmod N$ $\quad (z \in \mathbb{Z}_N^*)$

(... che x è un residuo quadratico mod N)

$\infty \rightarrow$ 

$\leftarrow$ poly 

Sceglie $z \in_R \mathbb{Z}_N^*$

$a = z^2 \bmod N$

$a \longrightarrow$

$\leftarrow b$

Sceglie $b \in_R \{0, 1\}$

Calcola

$r = z \cdot z^b$

$\longrightarrow r$

Se $r^2 = a \cdot x^b$, allora "SI"

altrimenti "NO"

Completezza: nota d

$$z^2 = (z \cdot a^b)^2$$

- $b = 0$: $z^2 \bmod N$ $\quad (a^b = a^0 = 1)$
- $b = 1$: $z^2 \cdot a^2 = z^2 x \bmod N$ $\quad (a^2 = x)$

Pertanto, il verificatore:

- se $b = 0$, calcola $z^2 = (z \cdot a^0)^2 = (z \cdot 1)^2 = z^2 = a \cdot x^0 = a \cdot 1 = a$

- se $b = 1$, calcola $z^2 = (z \cdot a^1)^2 = (z \cdot a)^2 = z^2 \cdot a^2 = z^2 \cdot x = a \cdot x$

Quindi, se x è un residuo quadratico, l'accetta sempre

Soundness: nota che

x x non è un residuo quadratico $\Rightarrow$ a e $a \cdot x$ non possono essere entrambi residui quadratici

2^2

Infatti, se $a = 2^2 \bmod N$ e risultasse anche

$a \cdot x = w^2 \bmod N$, allora si avrebbe

$x = w^2 \cdot (2^{-1})^2$, ovvero x sarebbe un residuo quadratico

(contrariamente all'ipotesi!)

Pertanto, il verificatore, indipendentemente dalla strategia del provatore, accetta con prob. $1/2$.

Infatti:

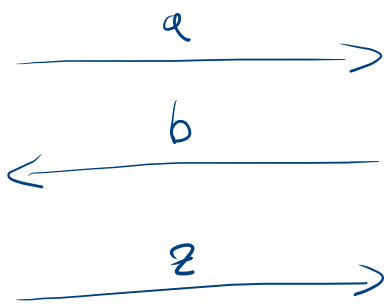
- se il provatore segue il protocollo, può rispondere correttamente solo se il verificatore sceglie $b = 0$

- se non lo segue, può rispondere comunque solo ad una delle sfide, e.g., sceglie a come non residuo allora può rispondere se $b = 1$ (mandando w tale che $w^2 = x \cdot a$) ma non alla sfida $b = 0$.

↳ (è un residuo quadratico, P è ∞, può trovare w)

Perfect Zero Knowledge.

Dobbiamo costruire il simulatore S^* per il transcript:



(transcript reale)

S^* (Alg. ppt) usa V^* come subroutine (black box)

Precisamente, procede come segue:

$S^*(x, N)$ — x: residuo quadratico mod N

Simulazione

1. Sceglie $z \in_R \mathbb{Z}_N^*$, $b \in_R \{0,1\}$
2. Pone $a = z^2 / x^b \bmod N$
3. Esegue $V^*(x)$
4. Invia a

(x)

subroutine di S^*

a →

V^*

← b'

sceglie $b' \in \{0,1\}$ (come vuole)

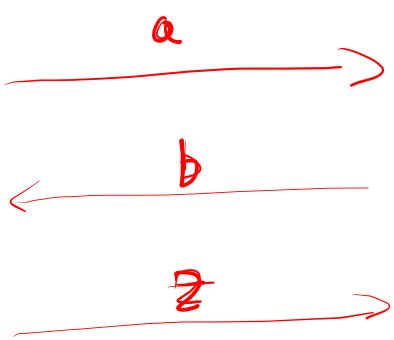
5. Se $b' = b$, dà in output (a, b, z);

altrimenti, ripete dal passo 1.

$\langle P, V^* \rangle (x, N) \equiv S^*(x, N)$

(transcript reale) (transcript simulato)

identicamente distribuiti



$a = z^2/x^b$ è un residuo quadratico uniforme perché z^2 è un residuo quadratico uniforme ed x è un residuo quadratico

b ha la stessa distribuzione dei bit b' generati dal V^* (un transcript viene generato nel quando S^* indovina b')

z per costruzione risulta $z^2 = a\, x^b$ (come nel transcript reale e con la stessa distribuzione)

Nota: la soundness può essere migliorata ripetendo il protocollo sequenzialmente k volte. OK!

Qualcuno potrebbe pensare di eseguirlo parallelamente:

(x) $\qquad x = a^2 \bmod N$

P $\qquad$ V

$a_1, a_2, \dots, a_n$ $\longrightarrow$

$z_1, \dots, z_n \in_R \mathbb{Z}_N^*$

$a_1 = z_1^2 \bmod N \dots a_n = z_n^2 \bmod N$

$b_1, b_2, \dots b_n \in_R \{0,1\}$

$b_1, b_2 \dots b_n$ $\longleftarrow$

$z_1 = z_1 \cdot a^{b_1}, \dots, z_n = z_n \cdot a^{b_n}$

$z_1, z_2, \dots, z_n$ $\longrightarrow$

Se $z_i = a_i \cdot x^{b_i} \bmod N$

per ogni $i = 1, \dots, n$

allora "SI".

Altrimenti "NO"

Parallelamente ...

- Completezza: OK
- Soundness: OK
- Zero Knowledge: ? non lo sappiamo ...

Nota: perché non possiamo "riprodurre" il simulatore di prima?

V^*

$b'_1 \dots b'_n$ ← scelgo $b'_1 \dots b'_n \in \{0,1\}$

Se $b_1 = b'_1 \dots b_n = b'_n$

S^* può indovinare gli n bit con prob $1/2^n$ (Occorrerebbero $\sim$ # esp. di pari e S^* è ppt)

Conclusione:

Non sappiamo costruire un simulatore e, allo stato attuale delle nostre conoscenze, a meno di risultati sorprendenti in teoria della complessità, non esiste.

Più in generale è improbabile che $\exists$ un sistema di prova a conoscenza zero perfetta con 3 round di comunicazione e probabilità di successo per un paratore malevolo trascurabile per il nostro problema

(... attenzione alla parallelizzazione!)

Altro esempio

Un sistema di prova a conoscenza zero per tuple DH

Un po' di background.

Schema di Commitment: per $s \in \mathbb{Z}_q$, (G, q, g, h) (pubblico)

$\lceil g^x, x \in \mathbb{Z}_q$ segreto

$com(s) = g^s h^z$, $z \in_R \mathbb{Z}_q$

$decom(c) = (s, z) \rightarrow c \overset{?}{=} g^s h^z$

Hiding: h^z è un elemento uniforme di G

$\Rightarrow$ s è protetto incondizionatamente

Binding: per aprire un commitment in modo diverso A dovrebbe trovare una coppia (s', z') tale che

$$g^s h^z = g^{s'} h^{z'}$$

$$\Rightarrow \quad g^{s - s'} = h^{z' - z}$$

$$\Rightarrow \quad g^{(s-s')(z'-z)^{-1}} = h^{(z'-z)(z'-z)^{-1}} = h^1$$

A sarebbe in grado di calcolare il log discreto di h

$$\Rightarrow \quad \log_g h = (s - s')(z' - z)^{-1} \quad \blacktriangle$$

Binding è garantita computazionalmente.

Sistema di prova a conoscenza zero

(G, q, g) (g^a, g^b, g^{ab})

A B C

P V

P → V: $com(s)$

V: $s \in_R \mathbb{Z}_q$

V: $com(s)$

P: $z \in_R \mathbb{Z}_q$

P: $y_1 = g^z$, $y_2 = B^z$

P → V: y_1, y_2

V → P: $decom(s)$

V: Open com

P: $r = z + as \bmod q$

P → V: r

V: $g^r = A^s y_1 \bmod q$

V: $B^r = C^s y_2 \bmod q$

Completeness : $g^z = g^{2+as} = g^{as} g^2 = A^s \cdot y_1 \bmod q$

$B^z = (B)^{2+as} = (B)^{as} \cdot B^2 = (g^b)^{as} \cdot B^2 = C^s \cdot y_2 \bmod q$

Soundness : Osserviamo che, quando P invia y_1 e y_2 non conosce s (protetto da com(s) incondizionatamente)

Se la tupla è (g^a, g^b, g^c), $a, b, c \in_R \mathbb{Z}_q$

l'unico modo che P ha di convincere V è "indovinare" y_1, y_2 e z tali che

$g^z = A^s y_1$ e $B^z = C^s y_2$

Ma risultano:

$$g^z = A^s y_1 = (g^a)^s y_1, \quad B^z = C^s y_2 \iff (g^b)^z = (g^c)^s \cdot y_2$$

$$g^z = g^{as+x} \qquad (g^x) \qquad g^{bz} = g^{cs+y}$$

Ma $z = as + x \Rightarrow b(as+x) = cs + y$

$\Rightarrow y = b(as+x) - cs$ (unico valore)

Pertanto, $\mathcal{P}$ dovrebbe indovinare s per calcolare i valori giusti, e ciò accade con prob. $1/2^q$

$S^*(G, q, g, (A, B, C))$

Simulazione

Sceglie $z \in_R \mathbb{Z}_q$

Esegue V^*

$com(s)$ ← V^*

Sceglie $y_1, y_2 \in_R G$

y_1, y_2 →

$decom(s)$ ←

V^* non si accorge che y_1 e y_2 sono casuali (nella realtà (y_1, B, y_2) sono una tripla DH)

Pone $y_1' = g^z / A^s$, $y_2' = B^z / C^s$

Dà in output:

per costruzione soddisfano le equazioni

Stessa distr. reale

$[com(s), y_1', y_2', decom(s), z]$

prodotti da V^* - identici al reale

$com(s)$ ←

y_1, y_2 →

$decom(s)$ ←

z →

(transcript reale)

Schemi per la condivisione di segreti

(Secret sharing scheme)

Polinomi su $\mathbb{Z}_p$

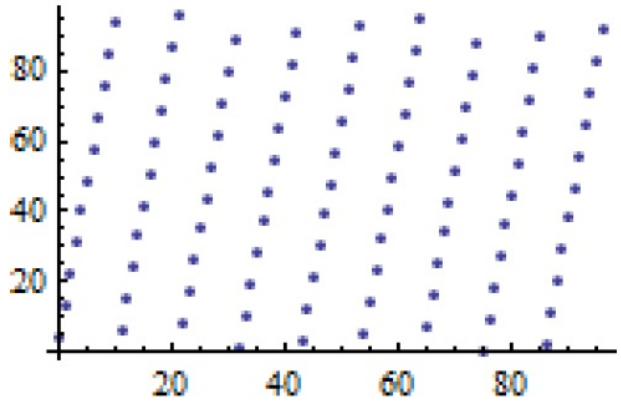


Grafico di una retta (mod 97)

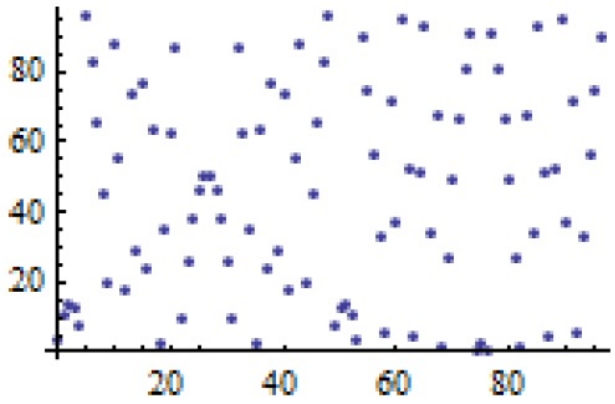


Grafico di una parabola (mod 97)

Matematicamente si comportano come i polinomi sui reali

Un polinomio di grado d ha al più d radici

Un polinomio di grado $d-1$ è univocamente determinato da d punti distinti $x_i \neq x_j$, $i \neq j$

$(x_0, f(x_0)) \quad \dots \quad (x_{d-1}, f(x_{d-1}))$

$f(x) = a_0 + a_1 x + \dots + a_{d-1} x^{d-1}$

Matrice di Vandermonde (Non singolare)

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^{d-1} \\ 1 & x_1 & \cdots & x_1^{d-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_{d-1} & \cdots & x_{d-1}^{d-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{d-1} \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_{d-1}) \end{pmatrix}$$

nota dati gli x_i

incognite

noti

$\exists !$ soluzione

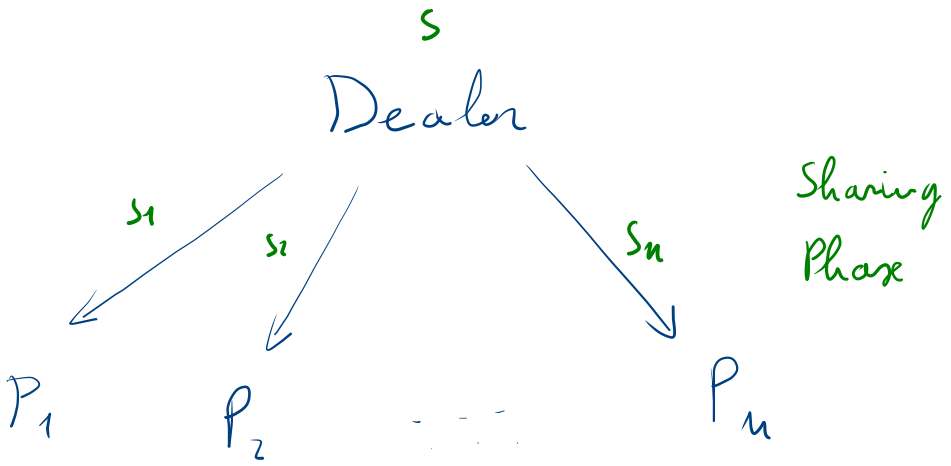
Come ricordate, il polinomio interpolante può essere anche calcolato più efficientemente ed espresso attraverso la formula di Lagrange

$$f(x) = \sum_{i=0}^{d-1} f(x_i) \cdot L_i(x)$$

dove $L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{d-1} \frac{(x - x_j)}{(x_i - x_j)}$ $\quad x_i \neq x_j, \ \forall i \neq j$

Nota: $L_i(x_i) = 1$ e $L_j(x_i) = 0 \quad \forall j \neq i$

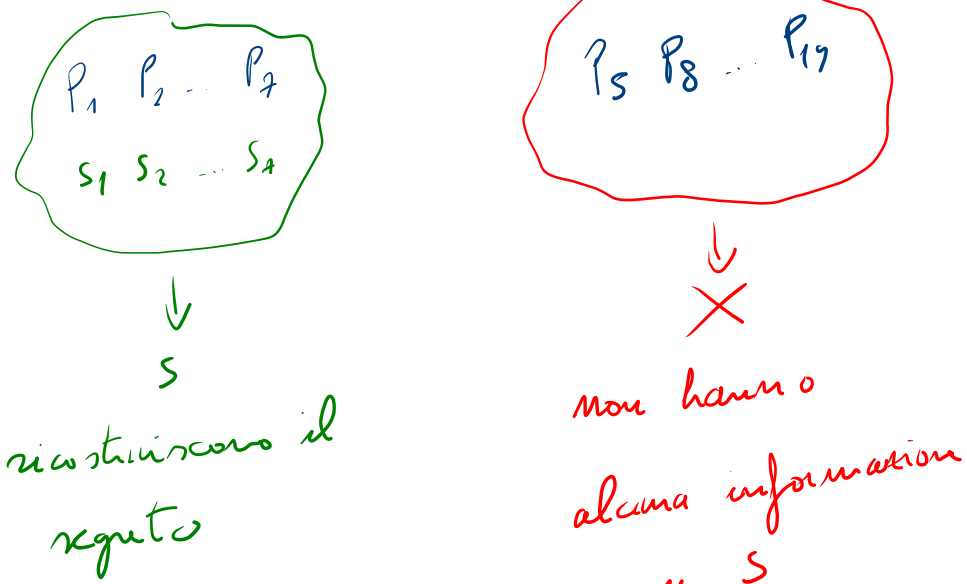
$\Rightarrow$ $f(x)$ vale esattamente $f(x_i)$ in ognuno degli x_i.



Il dealer vuole condividere un segreto S con gli n partecipanti in modo tale che

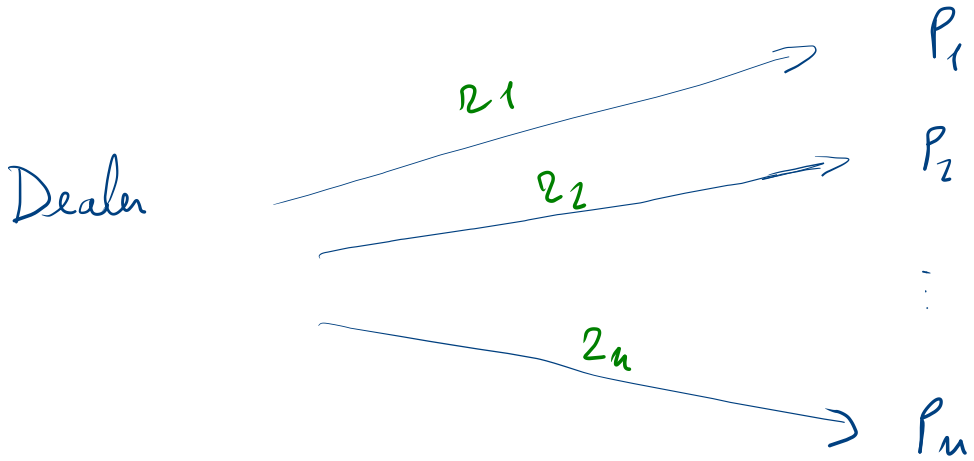
- sottoinsiemi qualificati ricostruiscano S
- sottoinsiemi proibiti non abbiano alcuna informazione su S

La Reconstruction Phase può essere condotta o da partecipanti o tramite un combiner



Warm up

$s \in \{0,1\}^n$ stringa di n bit



$r_i \in_R \{0,1\}^n$

$i = 1, \dots, n-1$

$r_n = r_1 \oplus \dots \oplus r_{n-1} \oplus s$

Quando tutti i partecipanti mettono assieme le proprie share

$$s = r_1 \oplus r_2 \oplus \dots \oplus r_{n-1} \oplus (r_1 \oplus r_2 \oplus \dots \oplus r_{n-1} \oplus s)$$

r_1: P_1, r_2: P_2, r_{n-1}: P_{n-1}, $(\dots)$: P_n

D'altra parte, qualsiasi sottoinsieme di partecipanti non ha alcuna informazione su S

- o i partecipanti dispongono di valori totalmente casuali, se P_u non è presente $\Rightarrow$ NO INFO

- oppure, se P_u è presente, manca almeno un 2_i per "rimuovere la maschera" ad 2_u e liberare S. Ma essendo $2_i \in_R \{0,1\}^u$, il segreto S può ancora essere qualsiasi valore in $\{0,1\}^u$ con prob uniforme. $\Rightarrow$ NO INFO

Nota: lo schema può essere facilmente riprodotto in altri gruppi al posto di $(\{0,1\}^n, \oplus)$

e.g. $(\mathbb{Z}_n, +_n)$

Nella forma più generale, gli insiemi qualificati e proibiti di partecipanti definiscono una "struttura d'accesso" al segreto

$$\mathcal{A} = (Q, F) \quad \text{su} \quad P = \{1, 2, \dots n\}$$

- $\mathcal{A}$: Struttura d'accesso
- Q: Famiglia di insiemi qualificati
- F: Famiglia di insiemi proibiti
- P: insieme di partecipanti

Le strutture d'accesso di interesse sono quelle monotone

$$A \in \mathcal{Q}, \quad A \subset B \Rightarrow B \in \mathcal{Q}$$

↑ I partecipanti in B possono sempre ricostruire usando solo quelli anche in A

Lo schema di Shamir

realizza una struttura d'accesso a soglia

$$\mathcal{Q} = \{ S \subseteq \{1, \dots, n\} : |S| \geq t \}$$

Ogni sottoinsieme di almeno t partecipanti ricostruisce

$$\mathcal{F} = \{ S \subseteq \{1, \dots, n\} : |S| < t \}$$

Ogni sottoinsieme di t-1 o meno partecipanti non ottiene alcuna informazione

Schema a soglia (t, n)

Sia $s \in \mathbb{Z}_p$ $(p$ primo, $p > n)$

$\longleftarrow$ devo essere almeno n punti distinti

Sharing Phase. Il Dealer sceglie unif. a caso un polinomio $a(x)$ di grado al più $t-1$, tale che $a(0) = s$. $\{ a_0 = s,\ a_i \in_R \mathbb{Z}_p$

Per $i = 1, \dots, n$, invia $s_i = a(i)$ al partecipante P_i

Reconstruction Phase. Ogni sottoinsieme di t partecipanti Q ricostruisce il segreto dalle proprie share, usando l'interpolazione di Lagrange

Nota: non ricostruisco $a(x)$. Calcolo direttamente il suo valore in 0, i.e., $a(0)$.

$$s = \sum_{i \in Q} s_i \cdot \lambda_{Q,i}$$

$$\lambda_{Q,i} = \prod_{j \in Q \setminus \{i\}} \frac{j}{j - i}$$

Correttezza. L'interpolazione di Lagrange garantisce che ogni sottoinsieme qualificato ricostruisca.

Sicurezza. Cosa garantisce che un sottoinsieme di al più $t-1$ partecipanti non ottiene alcuna info su $a(0)$?

Senza perdita di generalità, consideriamo $\{P_1 \dots P_{t-1}\}$

$a(0) = s_0 \longrightarrow$

$$\underbrace{\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 2^{t-1} \\ \vdots & & & \\ 1 & t-1 & & (t-1)^{t-1} \end{pmatrix}}_{t \times t} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{t-1} \end{pmatrix} = \begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_{t-1} \end{pmatrix}$$

Per ogni scelta di $s_0 \in \mathbb{Z}_p$ il sistema ammette un'unica soluzione

$\Downarrow$

Poiché i coefficienti $a_1, \dots a_{t-1}$ sono scelti unif. a caso $\Leftarrow$ ogni valore di s_0 è equamente probabile.

Osservazione: lo schema iniziale è uno schema a soglia (n, n)

Può essere esteso per realizzare uno (t, n)

- per ogni sottoinsieme di t partecipanti si usa uno schema (t, t) INDIPENDENTE DAGLI ALTRI

Inefficiente: P_i appartiene a $\binom{n-1}{t-1}$ sottoinsieme e riceve $\binom{n-1}{t-1}$ share, una per sottoinsieme

Shamir dà una share a P_i!

Lo schema iniziale è utile per realizzare però strutture d'accesso generali (quando non sappiamo far meglio...)