# 07 - Secret Sharing, Key Escrow

In molte situazioni una chiave o password è conosciuta solo a un singolo individuo.
**Escrow**: un accordo contrattuale in cui un terzo riceve ed eroga denaro o documenti per i principali soggetti operanti, con l'erogazione subordinata alle condizioni pattuite dalle parti negoziali
**Key Escrow**: un accordo in cui le chiavi private sono tenute "in escrow" tale che sotto certe circostanze una terza parte autorizzata può accedere per rivelare le chiavi.

## Secret sharing

- Si divide il segreto in $n$ parti e si inviano a diversi security officer (oppure si cripta il $i^{\text{th}}$ parte con la chiave pubblica del security officer $i$ e si tiene tutto sul proprio disco)
- Nessun sottogruppo di security officer dovrebbe essere in grado di ricostruire il segreto
- Solo tutti i $n$ security officer collaborando dovrebbero essere in grado di rivelare il segreto
- Problema: dividere $n$ in parti

### 2 way secret sharing

Si divida $S$ tra due security officer tale che:

- confidenziale: nessuno dei due security officer da solo possiede alcuna informazione su cosa sia $S$
- disponibilità: i due insieme possono ricostruire $S$
- Sia $S$ un segreto lungo $k$ bit, sia $n = 2$
- Si generi una stringa casuale $R$ di lunghezza $k$
- Si dia al primo security officer la stringa $S_1 = R$
- Si dia al secondo la stringa $S_2 = R \oplus S$
- Si ricostruisce il segreto come $S_1 \oplus S_2$

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Secret: | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | |
| Random string $R$: | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | $S_1$ |
| $\oplus$ | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | $S_2$ |
| $\oplus$ | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | Secret |

L'algoritmo è corretto:

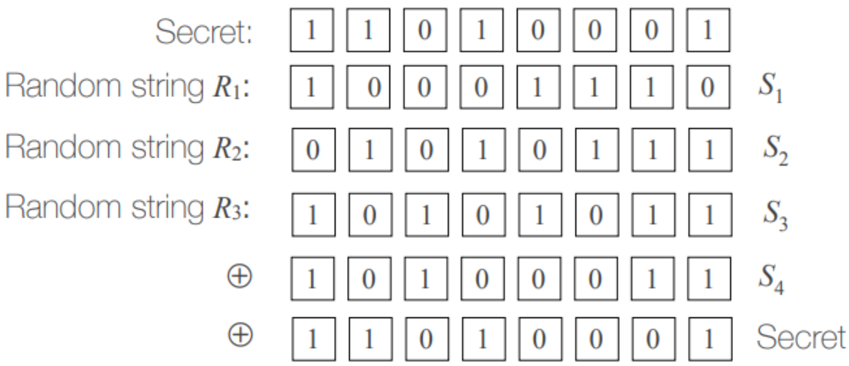
- confidenzialità: nessuno dei due security officer da solo può indovinare il segreto
- Disponibilità: i due lavorando insieme possono trovare il segreto $S$ come $S_1 \oplus S_2 = R \oplus (R \oplus S) = (R \oplus R) \oplus S = 0 \oplus S = S$

### N-way secret sharing

Generalizzazione a $n$

- si divida $S$ tra i $n$ security guard in modo da avere:
  - confidenzialità: nessun sottogruppo $m$ di security officers con $m < n$ ha informazioni riguardo $S$
  - disponibilità: tutti gli $n$ security officers insieme possono ricostruire $S$
- Sia $S$ un segreto lungo $k$ bit
- Genera $n-1$ random string $R_1, R_2, \dots R_{n-1}$ ognuno lungo $k$ bit
- Si dia ai primi $n-1$ security officers la stringa $R_1, R_2, \dots R_{n-1}$
- Si dia all'ultimo ($n^{\text{th}}$) security officer la stringa $R_1 \oplus R_2 \oplus \dots \oplus R_{n-1} \oplus R_n$

- Si ricostruisce il segreto $S_1 \oplus S_2 \oplus \ldots \oplus S_{n-1} \oplus S_n$




### Threshold schemes

Cosa succede se uno dei security officers muore?
è necessario ridurre i requirements per cui tutti le $n$ parti debbano essere presenti per ricostruire il segreto
$(t, n)$-Threshold scheme, dove $(t < n)$:

- segreto diviso in $n$ "shares"
- $t$ (o più) shares sono sufficienti per recuperare il segreto
- è impossibile recuperare il segreto con meno di $t$ shares

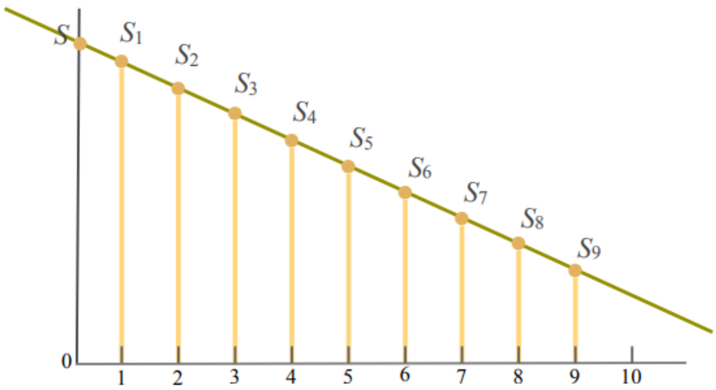
#### Shamir's method over a finite field

- Si generi un polinomio casuale di grado $t - 1$

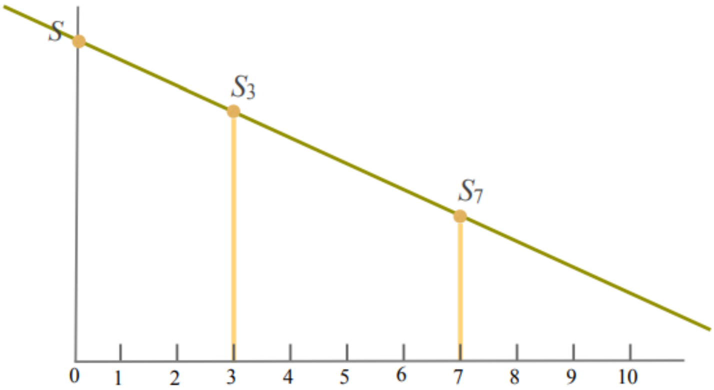
$$g(x) = (a_{t-1}x^{t-1} + at - 2x^{t-2} + \ldots + a_1x + S) \mod p$$

  dove i coefficienti $a_1 \ldots a_{t-1}$ sono selezionati casualmente, $S$ è il segreto e $p$ è un numero primo casuale (reso pubblico) più grande di qualsiasi dei coefficienti
- Si generino le $n$ shares come: $S_1 = g(1), S_2 = g(2), \ldots S_n = g(n)$
- Si scarti il polinomio (coefficiente e $S$)
- qualsiasi $t$ share è sufficiente per risolvere i $t$ parametri sconosciuti e ottenere il polinomio e quindi $S$

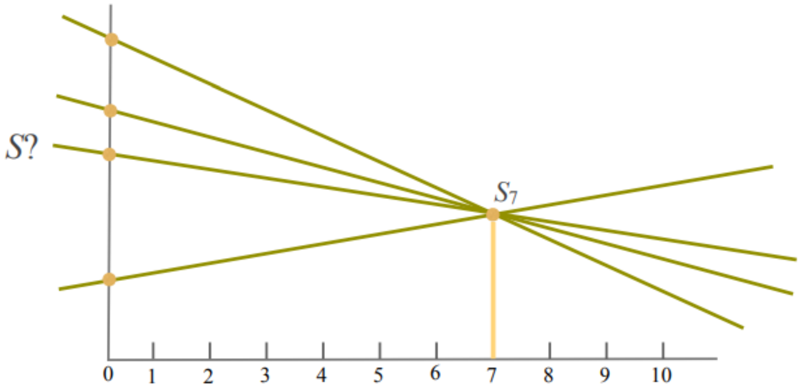
SI consideri un esempio di $(2, n)$ threshold scheme:
$g(x) = (ax + S) \mod p$




Date due o più shares:




Data una share:




Esempio numerico:

- $n = 3, t = 2$
- $S = 6, p = 17$
- Si generi un polinomio casuale di grado $t = 2 - 1 = 1$
  $g(x) = (4x + 6) \mod 17$
- Si calcoli le 3 shares da mandare ai 3 security officers:
  $g(1) = (4 + 6) \mod 17 = 10$
  $g(2) = (8 + 6) \mod 17 = 14$
  $g(1) = (12 + 6) \mod 17 = 1$
- per ricostruire il segreto bastano 2 security officers (es con 1 e 2)
- Sanno che il polinomio ha la forma $g(x) = (ax + S) \mod 17$
- Date le loro shares $g(1) = 10$ e $g(2) = 14$, hanno:
  - $g(1) = (a + S) \mod 17 = 10$
  - $g(2) = (a + S) \mod 17 = 14$
- risolvendo il sistema di equazioni per la $S$ sconosciutasi ottiene $S = 6$ che è il segreto iniziale