

Computer and Network Security

Lorenzo Rossi

December 28, 2021

Contents

I	Third Midterm	5
1	Secret Sharing	7
1.1	Trivial Secret Sharing	7
1.1.1	XOR Secret Sharing	7
1.1.2	Modular Secret Sharing	8
1.1.3	Shamir Secret Sharing	8

Part I

Third Midterm

Chapter 1

Secret Sharing

1.1 Trivial Secret Sharing

Supponiamo di avere un segreto e vogliamo dividerne la conoscenza in due persone (dette **shareholders**). Inoltre, vogliamo si viene a conoscenza del segreto se e solo se entrambe le parti rivelano la loro porzione di segreto. Chi



fornisce il segreto viene detto **dealer**, mentre chi riceve le porzioni del segreto sono detti **share**.

Nel caso in cui avessimo diviso il segreto in parti uguali, è una pessima idea poiché per indovinare il segreto abbiamo $\frac{1}{2^{N_{bit}}}$ probabilità di indovinare la password ed ora, avendo diviso il segreto in parti uguali, abbiamo una probabilità molto maggiore $\frac{1}{2^{\frac{N_{bit}}{2}}}$.

1.1.1 XOR Secret Sharing

Possiamo fare di meglio:

1. Prendi il segreto i.e. 0010.1101;
2. Genera una sequenza casuale **key** i.e. 1011.0100;
3. XOR il segreto e il valore casuale **one time pad** i.e. 1001.1001;
Fino ad ora abbiamo applicato un *Vernam cipher*.
4. Diamo ad uno share la sequenza casuale, mentre ad un altro diamo il valore dello XOR;
5. L'unione fra gli share dà la chiave.

Importante. *Il conoscere la chiave, cioè il valore casuale, non mi dà alcuna informazione riguardante la chiave. Lo stesso discorso vale per il valore dello XOR poiché, come dimostrato nel **perfect secrecy**, l'operatore di XOR tra una stringa pseudocasuale e un valore casuale non dà informazioni su quale sia la password. Questi due aspetti rappresentano un requisito di sicurezza.*

1.1.2 Modular Secret Sharing

Un altro possibile schema è quello di utilizzare le somme modulari:

1. Prendi il segreto S in bit, trasformato in digit i.e. $0010.1101 \rightarrow 45$;
2. Genera $RAND \bmod N$ i.e. $RAND \bmod 256 \rightarrow 180$;
3. Esegui $S - RAND \bmod N$ i.e. $S - RAND \bmod 256 \rightarrow 121$;

Importante. Questo schema è equivalente ad One Time Pad poiché abbiamo sommato un numero pseudocasuale con un numero casuale (in modulo). In altre parole, la probabilità di indovinare S conoscendo il valore casuale o il valore della somma è uguale alla probabilità di indovinare senza sapere nulla.

Questo metodo è più facile da implementare per essere condiviso con N shareholders. In particolare, genero 3 quantità truly random ed effettua la differenza tra il segreto e queste 3 quantità modulo N . Nel caso un attacker, riuscisse ad ottenere un numero sufficiente di share non può comunque ottenere la password, ma al più la differenza tra il segreto e le shares non prese.

Da qui è possibile definire il concetto di **perfect secrecy**: un avversario, conoscendo $n-1$ shares deve ancora possedere la probabilità di indovinare il segreto pari a quella di indovinare il segreto da zero.

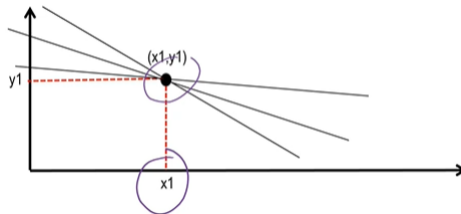
1.1.3 Shamir Secret Sharing

Fino ad ora abbiamo costruito uno schema detto (n,n) secret sharing scheme in cui il primo parametro è il numero delle persone necessarie a rilevare il segreto e il secondo parametro è il numero di parti: il segreto viene rilevato solo se tutte le n parti forniscono il segreto.

Un altro schema è (t,n) secret sharing scheme: il segreto è rilevato quando qualsiasi t delle n parti fornisce il segreto. Questo secondo problema è molto più complicato del trivial secret sharing.

Idea: Schema $(2,n)$

Il problema è quello di modellare uno schema per cui, conoscendo 2 degli n shareholders, posso ricostruire il segreto. Questo problema è riconducibile a quello di conoscere quanti punti sono necessari per definire una linea: ovviamente 2. Infatti conoscendo un solo punto (share) ho infinite rette passanti per quel punto e quindi è impossibile ricondurci



al segreto; tuttavia, conoscendo 2 punti (shares), tra essi passa solamente una sola retta e conseguentemente posso conoscere il segreto. Abbiamo comunque mantenuto la proprietà di poter avere un numero maggiore di 2 per ottenere il segreto, ma al minimo sono 2.

Procedura: Schema $(2,n)$

- **Dealer:** costruisce la linea:
 1. Coefficiente a : scelto casualmente;
 2. Segreto S : noto;

$$y = S + ax$$

Per esempio: $a = 15$ $S = 39$

- Distribuisce le shares ai n partecipanti scegliendo casualmente il valore x_i da introdurre nell'equazione della retta:
 - Shareholder 1: $x_1 = 1 \rightarrow share = (1, 54)$;
 - Shareholder 2: $x_2 = 2 \rightarrow share = (2, 69)$;
 - Shareholder 3: $x_3 = 3 \rightarrow share = (3, 84)$;
 - ...

Importante. La y viene calcolata in base alla funzione della retta; tuttavia, i punti degli shareholder sono mantenuti con (x,y) e il valore delle x_i possono essere noti a priori a patto che la y sia nascosta.

Procedura: Ricostruzione (2,n)

- Ricezione di due shares: $P_i = (x_i, y_i)$ $P_j = (x_j, y_j)$;
- Interpoli i punti per ricostruire l'equazione della retta:

$$\frac{y - y_i}{y_i - y_j} = \frac{x - x_i}{x_i - x_j}$$

Ottenendo:

$$y = y_i + \frac{x - x_i}{x_i - x_j}(y_i - y_j)$$

- Bisogna sostituire $x = 0$ per ottenere il segreto $y = S$;

Estensione al caso (t,n)

Estendendo il discorso precedentemente introdotto, ci si riconduce al caso di polinomi di grado $t-1$ unicamente definiti da t punti:

- Linea: 2 punti;
- Parabola (quadratic): 3 punti;
- Cubiche: 4 punti;
- ...

Generalizzazione: Schema (t,n)

- **Dealer:**
 1. Genera un polinomio casuale $p(x)$ di grado $t-1$;
 2. Imposta il segreto s come il termine noto del polinomio:

$$p(x) = s + a_1x + a_2x^2 + \dots + a_{t-2}x^{t-2} + a_{t-1}x^{t-1}$$

con s il segreto e i coefficienti delle x generati truly random;

3. Distribuisci uno share ad ogni shareholders:

$$(x_i, y_i) \rightarrow y_i = p(x_i)$$

- **Ricostruzione:** Collezione t shares su n disponibili e calcola il segreto utilizzando l'*Interpolazione di Lagrange* con $x = 0$:

$$s = \sum_{\text{shares } x_i} y_i \Lambda_{x_i} \quad \text{with} \quad \Lambda_{x_i} = \Lambda_{x_i}(0) = \prod_{\text{shares } x_k \neq x_j} \frac{-x_k}{x_i - x_k}$$

L'interpolazione di Lagrange si basa sul concetto che qualsiasi polinomio di grado $t-1$ con t punti noti, può essere decomposto come:

$$y = \sum_{i=1}^t y_i \Lambda_i(x)$$

In cui $\Lambda_i(x)$ è la base del polinomio calcolata come:

$$\Lambda_i(x) = \prod_{m=1, m \neq i}^t \frac{x - x_m}{x_i - x_m} \quad \Lambda_i(x_i) = 1; \quad \Lambda_i(x_m) = 0 \quad \text{for } m \neq i$$

Segretezza

Per discutere di quanto sia sicuro questo schema dobbiamo ricordare che in questo ambito la segretezza è così definita:

Finché si conoscono (t-1) shares non si dovrebbe avere nessuna informazione sul segreto che stiamo condividendo.

Lo schema di Shamir in questo senso non è sicuro poiché se conoscessi a priori il range in cui è compreso il segreto, potrei ciclare su uno share mancante per ottenere un segreto nel range voluto.

Esempio 1. Effettuiamo uno schema (3,4) in cui per conoscere il segreto dobbiamo conoscere almeno 3 share su 4. Dato che utilizziamo l'interpolazione di Lagrange il polinomio sarà di grado $t - 1$ e il termine noto sarà s :

$$y = 3x^2 + 52x + 32;$$

Abbiamo 4 shareholders, quindi dobbiamo generare 4 punti, generando un valore casuale x e sostituendolo nell'equazione precedente. Avendo posto rispettivamente i valori 1, 2, 3, 4, si ottengono i seguenti punti:

$$(1, 87), (2, 148), (3, 215), (4, 288)$$

Ora, occorre calcolare i valori di λ , supponendo di aver collezionato x_1, x_2, x_3 , come

$$\Lambda_i(x) = \prod_{m=1, m \neq i}^l \frac{x - x_m}{x_i - x_m} \quad \Lambda_i(x_i) = 1; \quad \Lambda_i(x_m) = 0 \quad \text{for } m \neq i$$

Ottenendo:

$$\Lambda_1(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)}$$

$$\Lambda_2(x) = \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)}$$

$$\Lambda_3(x) = \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}$$

Ora, per ricostruire il segreto occorre applicare

$$y = \sum_{i=1}^t y_i \Lambda_i(x)$$

Quindi:

$$s = y_1 \Lambda_{x_1}(0) + y_2 \Lambda_{x_2}(0) + y_3 \Lambda_{x_3}(0) = 87(30) + 148(-3) + 215(1) = 32$$

Ora supponiamo di non sapere uno share (d) e vogliamo verificare se questo schema garantisce secrecy o meno. Sostituendo imponiamo:

$$s = y_1 \Lambda_{x_1}(0) + d \Lambda_{x_2}(0) + y_3 \Lambda_{x_3}(0) = 476 - 3d$$

Ipotizziamo che il range in cui vive s è noto e compreso tra 0 e 100. Possiamo indovinare il segreto? Sì, basta ciclare sulle d :

- Con $d = 125 \rightarrow s = 101$;
- Con $d = 126 \rightarrow s = 98$;
- Con $d = 127 \rightarrow s = 95$;
- Da varie prove si capisce che d è nel range $126 \leq d \leq 158$;

Quindi, conoscere 2 su 3 in uno schema 3 su 4 ci permette di escludere tutti i valori d non ammissibili.

Real Shamir Secret Sharing

Lo schema reale utilizza l'aritmetica modulare (con p numero primo) invece di quella reale e le operazioni effettuate sia con il segreto sia con il polinomio devono essere scelti nel campo dei numeri primi.

L'interpolazione rimane uguale.

Importante. La regola per scegliere il numero primo p deve essere più grande del dominio del segreto per avere un segreto uniformemente distribuito e non è necessario che sia grande.