Corso di Crittografia

Prof. Dario Catalano

Fully Homomorphic Ecryption

Statistical Indistingushability

Date due distribuzioni di probabilità X, Y definite su un supporto (finito) A.

$$\Delta_{X,Y} = \sum_{\omega \in A} |\Pr[X = \omega] - \Pr[Y = \omega]|$$

- X e Y sono statisticamente indistinguibili se $\Delta_{X,Y}$ è una funzione trascurabile in $|A|$

(Fully) Homomorphic Encryption

Costituito da **quattro** algoritmi

$$(\textbf{KeyGen}, \textbf{Eval}, \textbf{Enc}, \textbf{Dec})$$

.

- **KeyGen**, **Enc**, **Dec** sono definiti nel solito modo.
- **Eval** prende in input una funzione f e n crittotesti $C_i = \textbf{Enc}_{pk}(m_i)$ $(i = 1, .., n)$ e calcola $C_f = \textbf{Enc}_{pk}(f(m_1, ..., m_n))$

(Fully) Homomorphic Encryption

- Faremo vedere una soluzione in cui lo spazio dei messaggi è $\{0,1\}$.
- Il nostro schema supporta addizioni e moltiplicazioni modulo 2.
- Add (mod 2) implementa lo $\oplus$, Mult (mod 2) implementa l'operazione $\wedge$.
- Ciò è sufficiente per valutare ogni circuito booleano.

$$\neg x \rightarrow x \oplus 1^{\ell} \quad (1)$$

$$x \vee y \rightarrow (x \wedge \neg y) \oplus (\neg x \wedge y) \quad (2)$$

(ℓ lunghezza di x)

Strong Homomorphic Encryption

FHE$=$ ($\mathbf{KeyGen}, \mathbf{Eval}, \mathbf{Enc}, \mathbf{Dec}$) FHE con spazio dei messaggi $\{0,1\}$. FHE è omomorfico in senso forte se

- ($\mathbf{KeyGen}, \mathbf{Enc}, \mathbf{Dec}$) è un cifrario (asimmetrico) sicuro in senso ind-cpa.
- Per ogni coppia di bit $b, b' \in \{0,1\}$ e $\forall (pk, sk) \leftarrow \mathbf{KeyGen}$, $c \leftarrow \mathbf{Enc}_{pk}(b)$, $c' \leftarrow \mathbf{Enc}_{pk}(b')$ si ha (informalmente)
  1. $c^* \leftarrow \mathrm{Add}_{pk}(c, c')$ è statist. indisting. da $c^* \leftarrow \mathbf{Enc}_{pk}(b \oplus b')$
  2. $c^* \leftarrow \mathrm{Mult}_{pk}(c, c')$ è statist. indisting. da $c^* \leftarrow \mathbf{Enc}_{pk}(b \wedge b')$

Public vs Private Key

- Ogni cifrario (omomorfico) simmetrico può essere trasformato in un cifrario asimmetrico con le **stesse** proprietà.
- Dato $\mathrm{FHE}_{SE} = (\mathbf{KeyGen}, \mathbf{Enc}, \mathbf{Eval}, \mathbf{Dec})$ lo trasformiamo in un cifrario a chiave pubblica nel seguente modo.
- La chiave privata è la chiave segreta dello schema iniziale.
- La chiave pubblica è (una) cifratura di 0 e una cifratura di 1.
- Per cifrare un nuovo messaggio b basta utilizzare la chiave pubblica insieme a **Eval**

Learning Divisor with Noise (LDN) Assumption

- P primo (random) di n bit,
- Q primo (random) di n^4 bit, $\mathbf{E}$ (intero arbitrario) $\mathbf{E} \geq 2^{n^{0.1}}$, $N = PQ$ pubblico.

Nessun A polinomialmente limitato può distinguere le seguenti distribuzioni di probabilità, meglio che a caso.

1. $X_1, \ldots, X_t \in_R \mathbb{Z}_N$
2. For $i = 1 \ldots t$

$$R_i \leftarrow_R \mathbb{Z}_Q; E_i \leftarrow_R [-\mathbf{E}, +\mathbf{E}]$$
$$X_i \leftarrow R_i P + 2E_i$$

t polinomiale in n

Lo schema di base

KeyGen P primo (random) di n bit, Q primo (random) di n^4 bit, $N = PQ$ chiave pubblica, P chiave privata.

Enc(b) Output $\mathbf{Enc}^{2^{\sqrt{n}}}_{N,P}(b)$ dove $\mathbf{Enc}^{\mathbf{E}}_{N,P}(b)$ è definito come segue

1. $R \leftarrow \mathbb{Z}_Q$, $E \leftarrow [-\mathbf{E}, +\mathbf{E}]$
2. Output $RP + 2E + b \bmod N$

Dec(X) Output $X - \lceil X/P \rfloor P \bmod 2$

Omomorfismo

- $\text{Add}_N(X, X') = X + X' \bmod N$
- $\text{Mult}_N(X, X') = X \cdot X' \bmod N$

Sia $\mathcal{E}^{\mathbf{E}}_{N,P}(b)$ il seguente insieme

$$\{X \ : \ X = RP + 2E + b \bmod N,\ R \in_R \mathbb{Z}_Q,\ E \in [-\mathbf{E}, +\mathbf{E}]\}$$

- Se $E' \geq E$, $\mathcal{E}^{\mathbf{E}}_{N,P}(b) \subseteq \mathcal{E}^{\mathbf{E}'}_{N,P}(b)$
- Se E ed e' sono abbastanza "distanti" da P i due insiemi $\mathcal{E}^{\mathbf{E}}_{N,P}(0)$ e $\mathcal{E}^{\mathbf{E}}_{N,P}(1)$ sono disgiunti
- Nel nostro schema $\mathbf{E} = 2^{\sqrt{n}}$

Omomorfismo (cont.)

Affinché il nostro schema possa essere fully homomorphic si dovrebbe avere

- Se $X \in \mathcal{E}^{\mathbf{E}}_{N,P}(b)$ e $X' \in \mathcal{E}^{\mathbf{E}}_{N,P}(b')$

$$\mathrm{Add_N}(X, X') \in \mathcal{E}^{\mathbf{E}}_{N,P}(b \oplus b')$$

- $\mathrm{Add_N}(X, X')$ uniformemente distribuito in $\mathcal{E}^{\mathbf{E}}_{N,P}(b \oplus b')$

Nessuna di queste condizioni è rispettata!

Omomorfismo (cont.)

Lemma

$\forall\, \mathbf{E}, \mathbf{E}'$ *se* $X \in \mathcal{E}^{\mathbf{E}}_{N,P}(b)$ *e* $X' \in \mathcal{E}^{\mathbf{E}}_{N,P}(b')$ *allora*

1. $\mathrm{Add}_{\mathrm{N}}(X, X') \in \mathcal{E}^{\mathbf{E}+\mathbf{E}'}_{N,P}(b \oplus b')$
2. $\mathrm{Mult}_{\mathrm{N}}(X, X') \in \mathcal{E}^{\mathbf{3E}\cdot\mathbf{E}'}_{N,P}(b \wedge b')$

Se partiamo da crittotesti con rumore $\mathbf{E} = 2^{\sqrt{n}}$ possiamo effettuare un numero limitato di addizioni e sottrazioni

Problema: Come tenere traccia delle crescita del rumore?

Circuiti Aritmetici

1. C prende in input interi e produce un intero in output.
2. C si compone solo di gates additivi, moltiplicativi (contiene anche le costanti 1 e 0)
3. $\mathcal{P}(C) : \mathbb{Z}^m \to Z$ polinomio computato da C
4. Se C è un circuito booleano $C(b_1, \ldots, b_m) = \mathcal{P}(b_1, \ldots, b_m) \bmod 2$

Sia $f : \mathbb{Z}^m \to Z$ polinomio e $M \geq 0$, definiamo $|f|_M$

$$max_X = \{|f(x_1, \ldots, x_m)|\}_{X=\{x_1,\ldots,x_m\}, \forall\, i\, |x_i|<M}$$

Se f ha grado d ed i suoi monomi sono tutti di magnitudo al massimo e, allora

$$|f|_M \leq eM^d \binom{m+d}{d} \leq cM^d m^{2d}$$

Lo schema di base

KeyGen P primo (random) di n bit, Q primo (random) di n^4 bit, $N = PQ$ chiave pubblica, P chiave privata.

Enc(b) Output $\mathbf{Enc}^{2^{\sqrt{n}}}_{N,P}(b)$ dove $\mathbf{Enc}^{\mathbf{E}}_{N,P}(b)$ è definito come segue

1. $R \leftarrow \mathbb{Z}_Q$, $E \leftarrow [-\mathbf{E}, +\mathbf{E}]$
2. Output $RP + 2E + b \bmod N$

Dec(X) Output $X - \lceil X/P \rfloor P \bmod 2$

Somewhat Homomorphic Encryption

Lemma

Sia C un circuito aritmetico tale che

$$|\mathcal{P}(C)|_{2\mathbf{E}+1} < P/10$$

Per ogni $b_1, \ldots, b_m \in \{0,1\}$ sia $X_i \leftarrow \mathbf{Enc}^{\mathbf{E}}_{N,P}(b_i)$. Supponiamo inoltre di valutare C sugli X_i (utilizzando $\mathrm{Add_N}$ e $\mathrm{Mult_N}$) ottenendo X^* come risultato. Allora

$$\mathbf{Dec}_{N,P}(X^*) = C(b_1, \ldots, b_n)$$

Verso uno schema Fully Homomorphic

Supponiamo di avere due procedure (magiche!)

Clean Prende in input N, $X \in \mathcal{E}^{2^{n^{0.9}}}_{N,P}(b)$ e restituisce $X' \in \mathcal{E}^{2^{n^{0.3}}}_{N,P}(b)$

ReRand Prende in input N, $X \in \mathcal{E}^{2^{n^{0.4}}}_{N,P}(b)$ e restituisce $X' \in \mathcal{E}^{2^{\sqrt{n}}}_{N,P}(b)$

Con queste due procedure lo schema di base diverrebbe FH!

- $\text{Add}_N(X, X') = \mathit{ReRand}_N(\mathit{Clean}_N(X + X' \bmod N)$
- $\text{Mult}_N(X, X') = \mathit{ReRand}_N(\mathit{Clean}_N(X \cdot X' \bmod N)$

Costruire Clean: Bootstrappable Ecryption

Supponiamo che **Dec** abbia un circuito molto semplice

$$|\mathcal{P}(C_{\mathbf{Dec}})|_{2^{n^{0.1}}} < 2^{n^{0.3}}$$

Modifichiamo lo schema di base e costruiamo Clean come segue

1. Siano $Y_1, \ldots Y_n$ elementi della chiave pubblica, dove

$$Y_i = \mathbf{Enc}_{N,P}^{2^{n^{0.1}}}(P_i)$$

   ($P = P_1 \cdots P_n$ chiave segreta)
2. Siano $Y_{n+1}, \ldots Y_m$ ($m = n + n^5$) i bit di X

$$Y_{n+1} = X_1, \ldots, Y_m = X_{n^5}$$

Osservazione: Ogni bit di X corrisponde ad una cifratura di se stesso (con errore 0)

Problemi

1. Il circuito che implementa **Dec** non è piccolo! (Tutt'altro!)
   - In $X - \lceil X/P \rfloor P \bmod 2$ la parte "costosa" è calcolare $\lceil X/P \rfloor$
2. Avere una funzione di decifratura troppo semplice (assimilabile ad un polinomio di grado piccolo) è pericoloso
   - "Apprendere" low-degree polynomials richiede $\binom{n}{d}$ passi.
   - Per $d = n^{0.1}$, tuttavia, ciò non è fattibile in modo efficiente

Squashing

Idea: Pubblichiamo (in modo sicuro) un valore prossimo a $1/P$

- Sia $M = 2^{100n}$, $\alpha_1, \ldots, \alpha_{m'} \in_R [M]$ ($m' =$poly(m))
- Sia T un sottoinsieme (random) di indici in $[m']$, $|T| = n^{1/100}$ tale che

$$\sum_{i \in T} \alpha_i = \lceil M/P \rfloor \bmod M$$

- Per ogni $i \in [m']$ sia

$$t_i = \begin{cases} 1 & i \in T \\ 0 & i \notin T \end{cases}$$

- Sia $T_i = \mathbf{Enc}_P^{2^{n^{0.1}}}(t_i)$
- Aggiungiamo alla PK $(M, \alpha_1, \ldots, \alpha_{m'}, T_1, \ldots, T_{m'})$

Sparse Subset Sum (SSS) Assumption

Per ogni $m, \epsilon > 0$ e $\beta \in [M = 2^m]$ supponiamo che esista un $m' =$poly(m) tale che le seguenti distribuzioni sia (computazionalmente) indistinguibili.

1. $\alpha_1, \ldots, \alpha_{m'} \in_R [M]$
2. Sia T un sottoinsieme (random) di indici in $[m']$, $|T| = m^\epsilon$. Per $i \notin T$, $\alpha_i \in_R [M]$.

$$\sum_{i \in T} \alpha_i = \beta \bmod M$$

La procedura Clean

Su input X (pari)

1. For $i = 1, \ldots, m'$ compute $\tilde{\alpha}_i = X \cdot \alpha_i$
2. Construct a (mod 2) circuit $C(t_1, \ldots, t_{m'})$ to compute

$$C(t_1, \ldots, t_{m'}) = \left\lceil \frac{\sum_{i=1}^{m'} \tilde{\alpha}_i \cdot t_i}{M} \right\rfloor \bmod 2$$

3. Invoke C on input the ciphertexts $T_1, \ldots, T_{m'}$ to get the ciphertext X'
   - Using basic $\mathrm{Add_N}$ and $\mathrm{Mult_N}$