PRIVACY-PRESERVING INFORMATION SHARING

Carlo Blundo
DISA-MIS, Università degli Studi di Salerno

De Cifris Athesis, Università degli Studi di Trento, Dipartimento di Matematica
10 settembre 2019

CONDIVIDERE INFORMAZIONI E PRIVACY

- Necessario quando entità con limitata mutua fiducia vogliono o devono condividere delle informazioni
  - Durante il processo deve essere divulgata solo la quantità minima di informazione richiesta

IL PROBLEMA DEI MILIONARI

- Due milionari, Alice e Bob, sono a cena
- Il più ricco pagherà il conto
- Possono sapere chi di loro è più ricco senza rivelare la loro vera ricchezza?

- **Problema risolto da Andrew Yao nel 1982**

SECURE TWO PARTY COMPUTATION

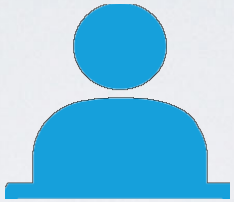
Alice: a

Bob: b

Mondo Ideale

SECURE TWO PARTY COMPUTATION

Terza parte fidata



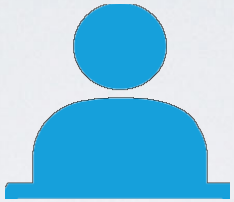
Alice: a

Bob: b

Mondo Ideale

SECURE TWO PARTY COMPUTATION

Terza parte fidata



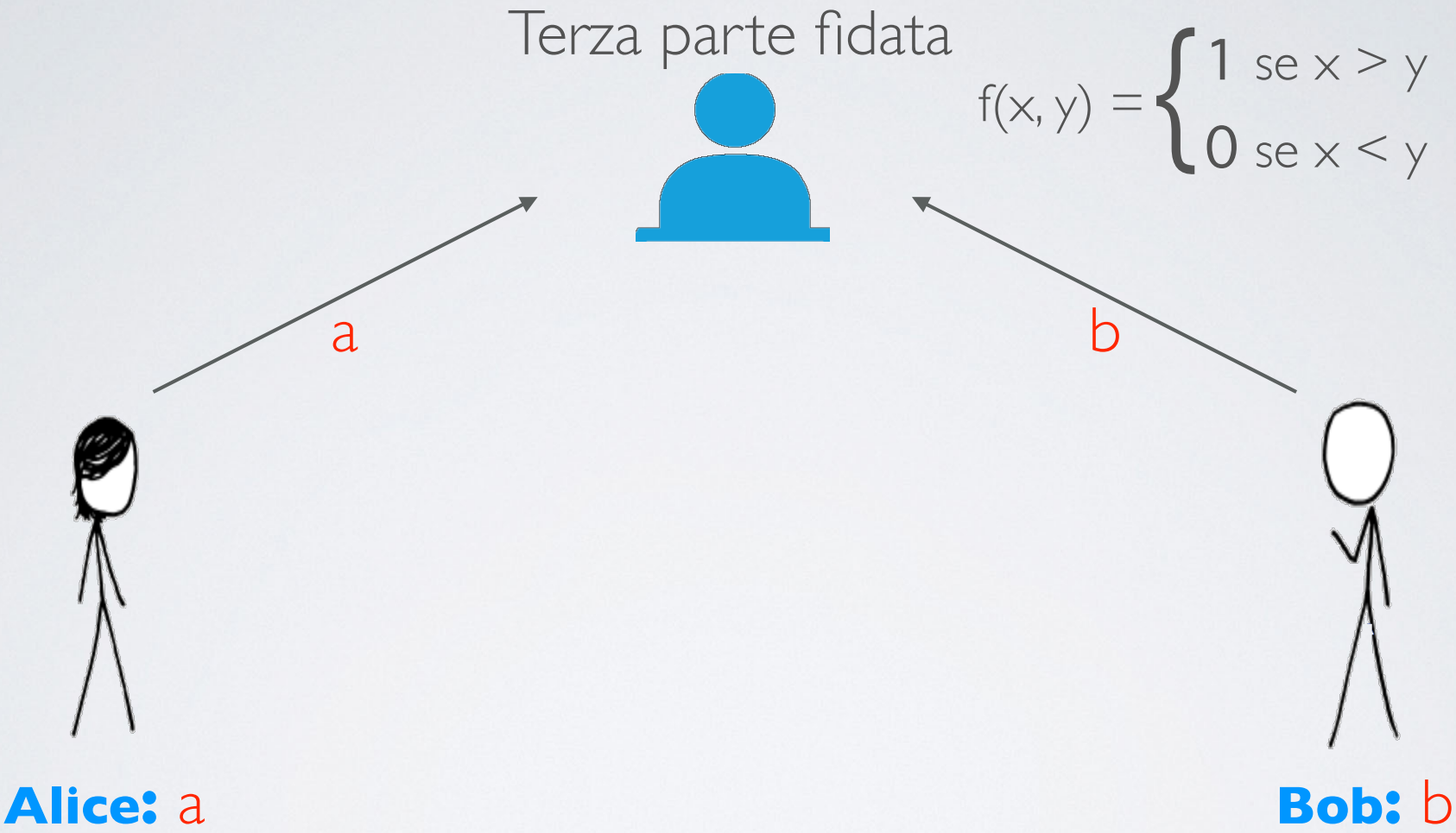
$$f(x, y) = \begin{cases} 1 \text{ se } x > y \\ 0 \text{ se } x < y \end{cases}$$

Alice: a

Bob: b

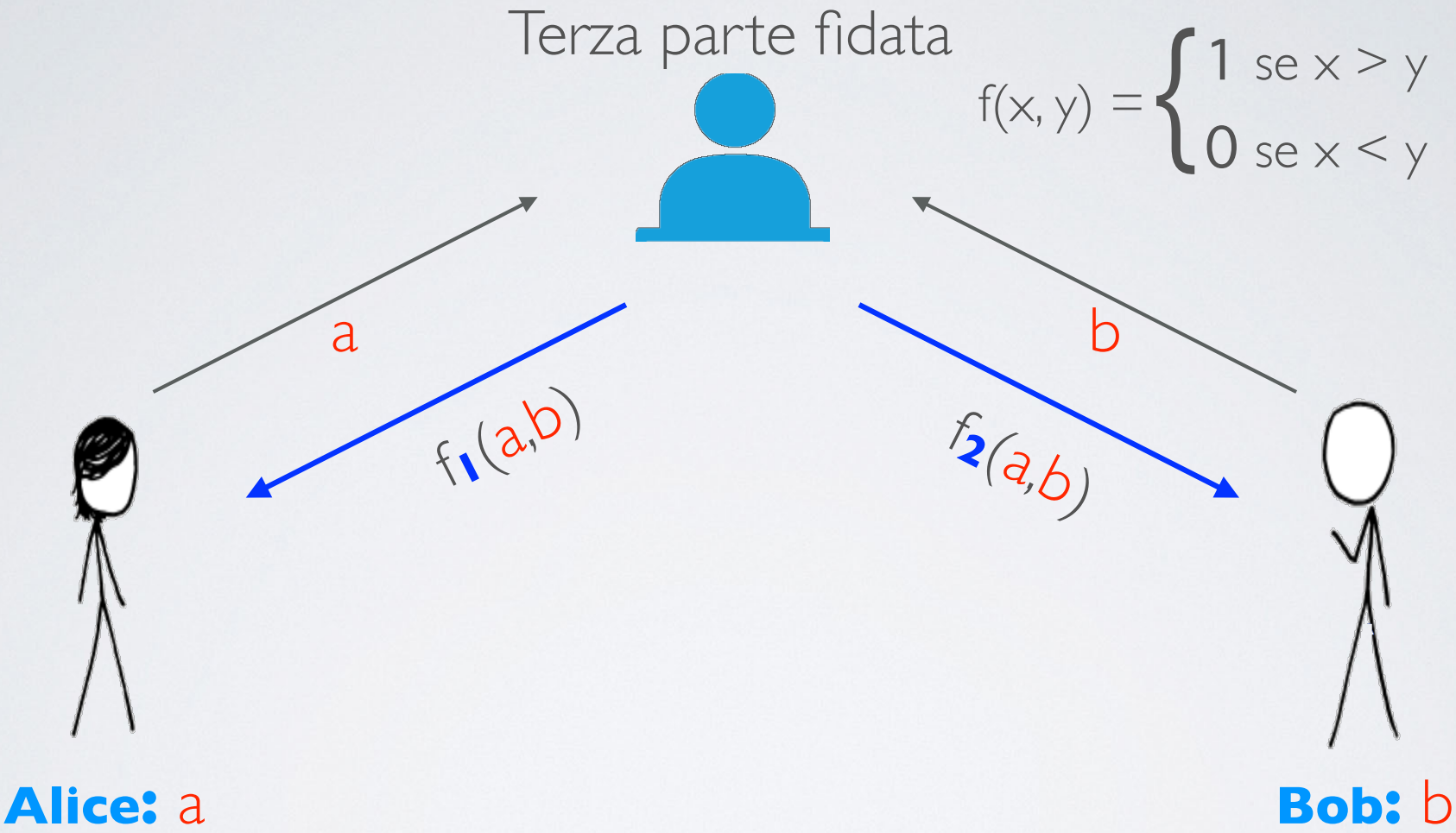
Mondo Ideale

SECURE TWO PARTY COMPUTATION



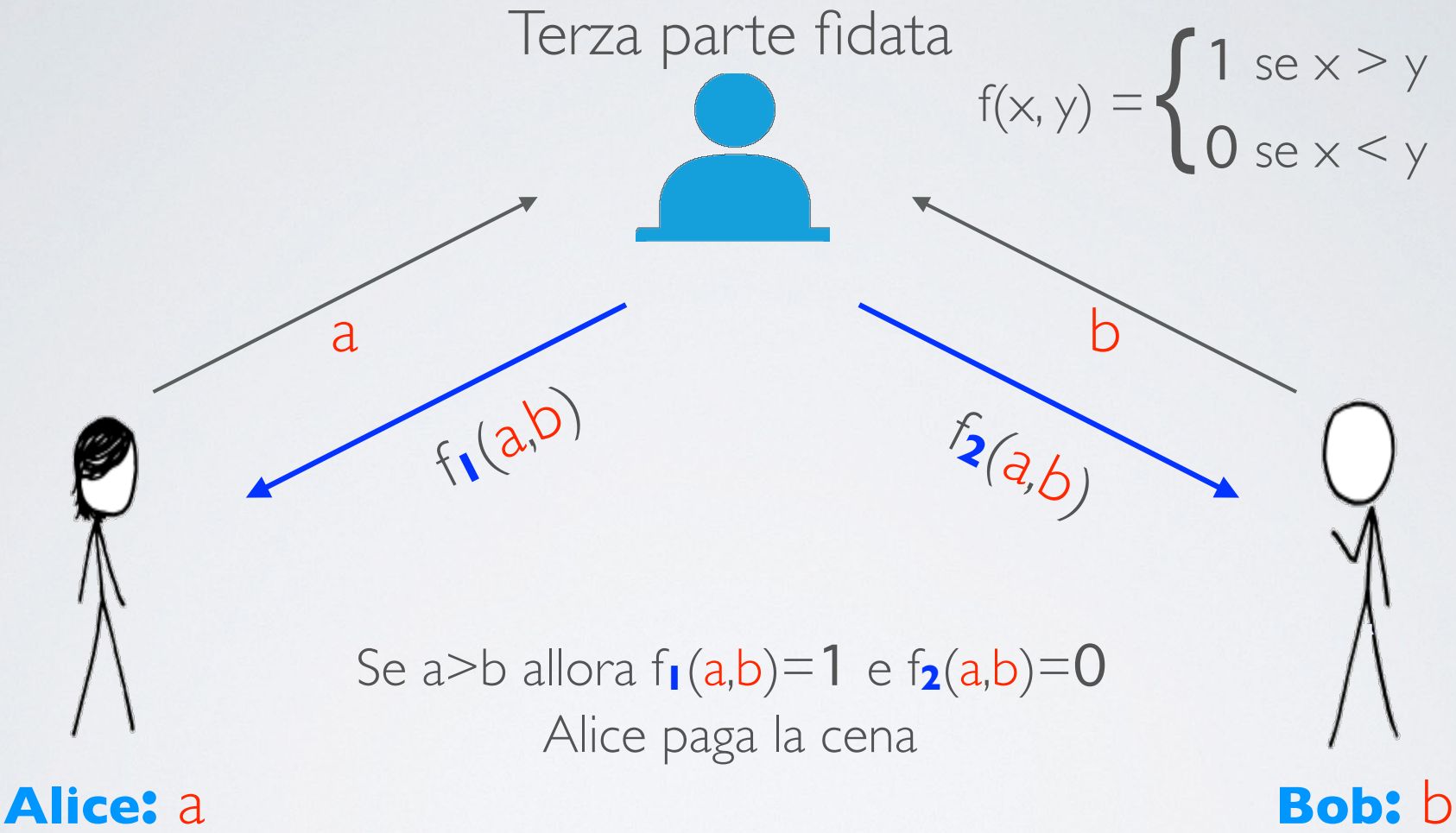
Mondo Ideale

SECURE TWO PARTY COMPUTATION



Mondo Ideale

SECURE TWO PARTY COMPUTATION



Mondo Ideale

SECURE TWO PARTY COMPUTATION

Alice: a

Bob: b

Mondo Reale

SECURE TWO PARTY COMPUTATION

Mondo Reale

SECURE TWO PARTY COMPUTATION

Mondo Reale

Mondo Ideale ≈ *Mondo Reale*

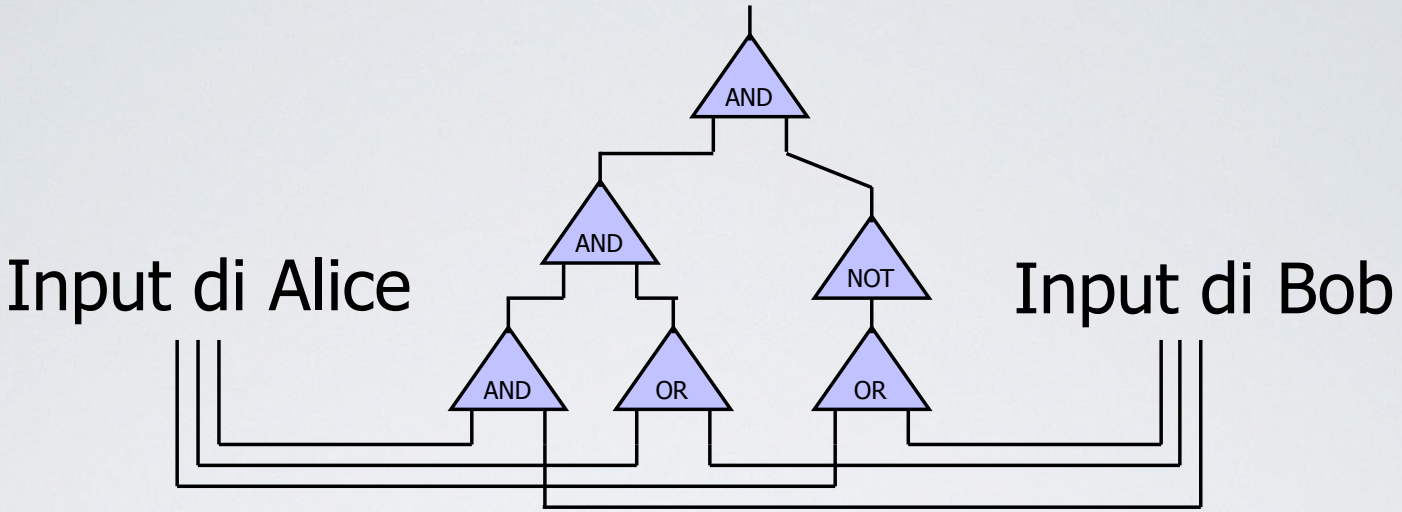
- **Correttezza**
  - L'output del protocollo è identico a quello della funzionalità che deve essere calcolata
- **Privatezza**
  - Alla fine del protocollo Alice e Bob apprendono solo il valore del proprio output e tutto ciò quello che può essere dedotto da esso

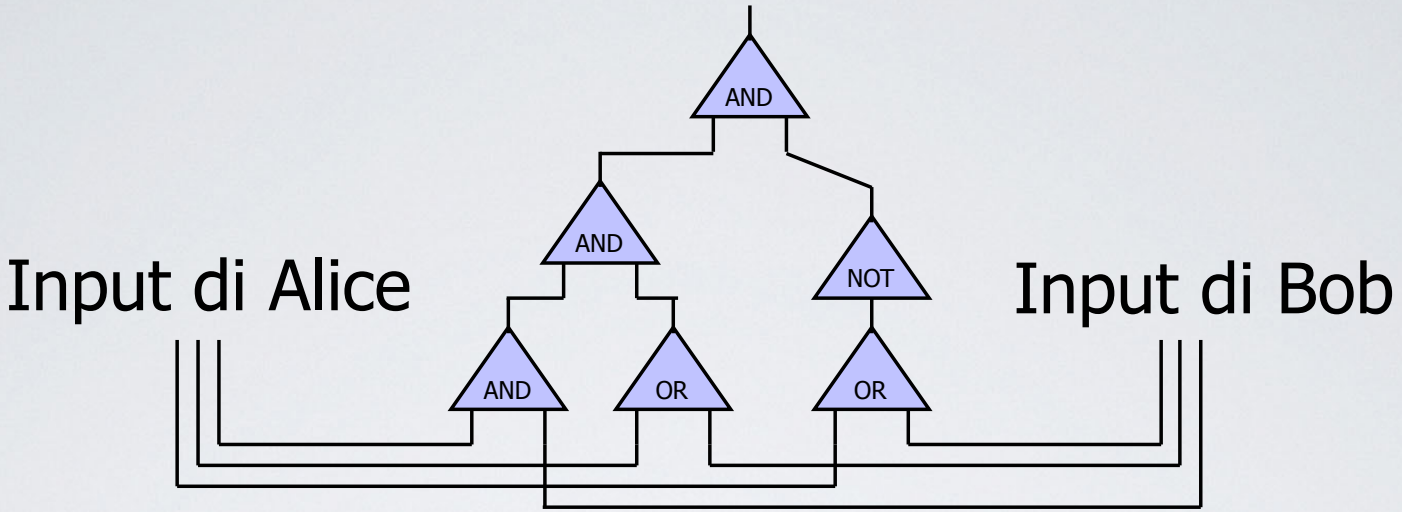
TIPI DI AVVERSARI

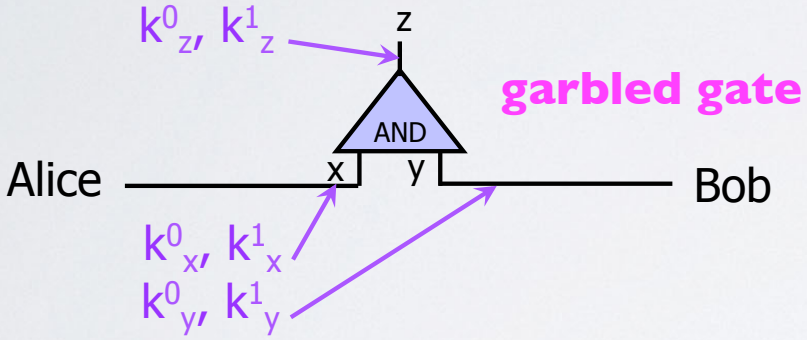
- **Honest-but-Curious** (semi-honest)
  - Hones: segue il protocollo non modifica gli input
  - Curious: tenta di inferire informazioni sull'input dell'altro giocatore
- **Malicious**
  - Devia in maniera arbitraria dal protocollo

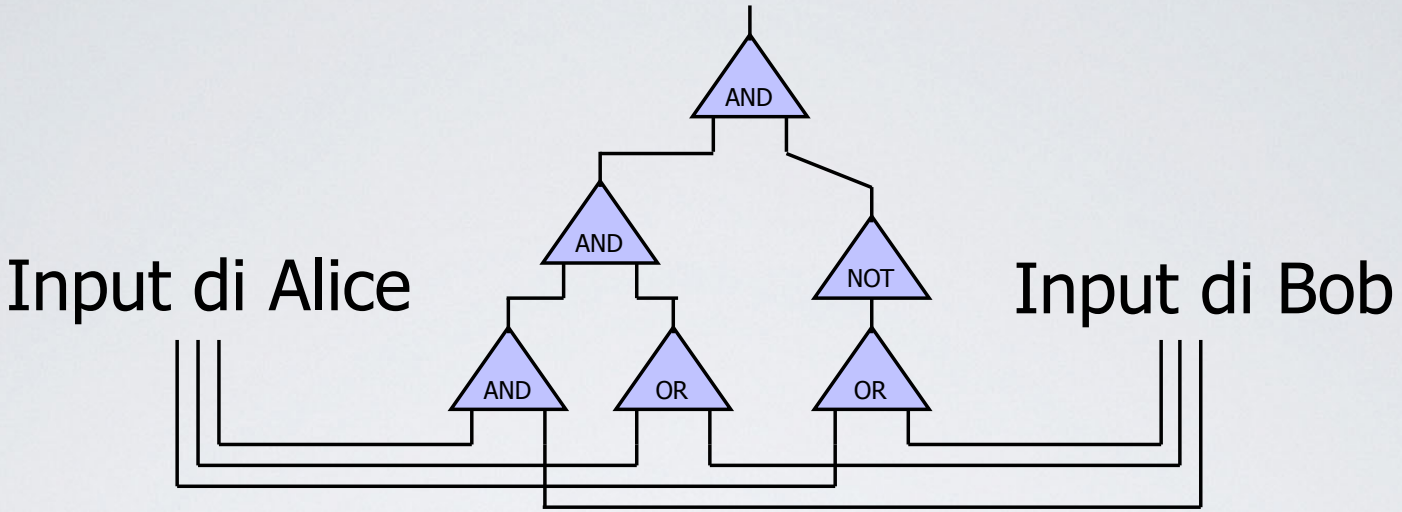
PROTOCOLLI

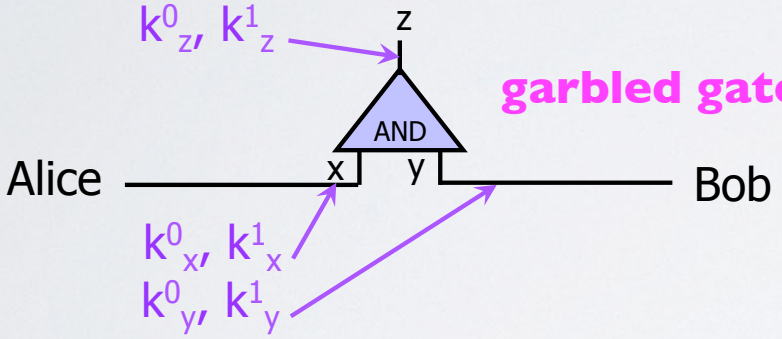
- **Generici**
  - Funzione rappresentata tramite un circuito (offuscato), Bob prepara il circuito e gli input (offuscati) e Alice valuta il circuito
- **Ad-hoc**
  - Implementano una funzione specifica (e solo quella) sono, in genere, basati su crittografia a chiave pubblica


AND
AND
NOT
Input di Alice
Input di Bob
AND
OR
OR


AND
AND
NOT
Input di Alice
Input di Bob
AND
OR
OR


k^0_z, k^1_z
z
garbled gate
AND
Alice
x
y
Bob
k^0_x, k^1_x
k^0_y, k^1_y

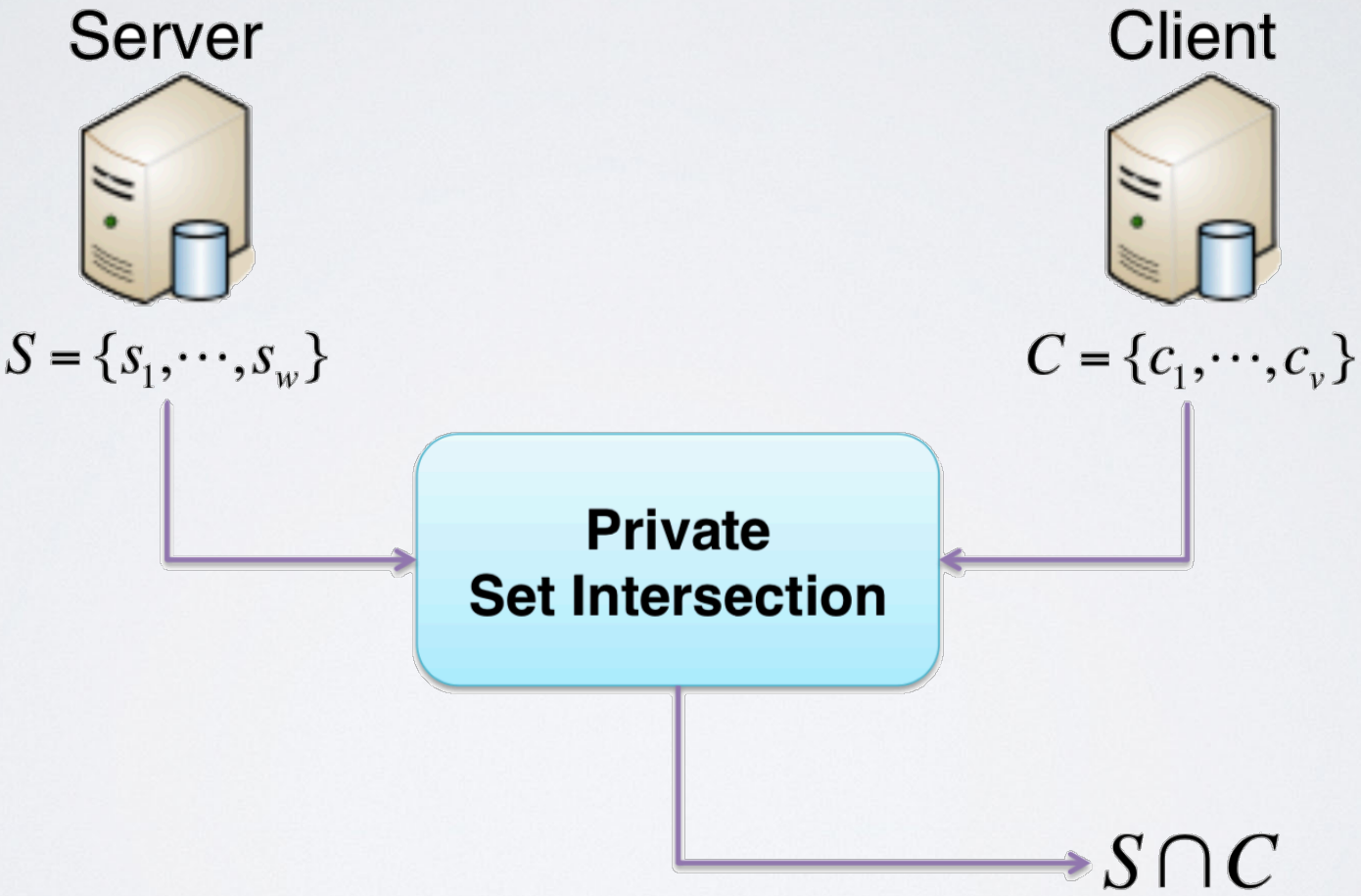




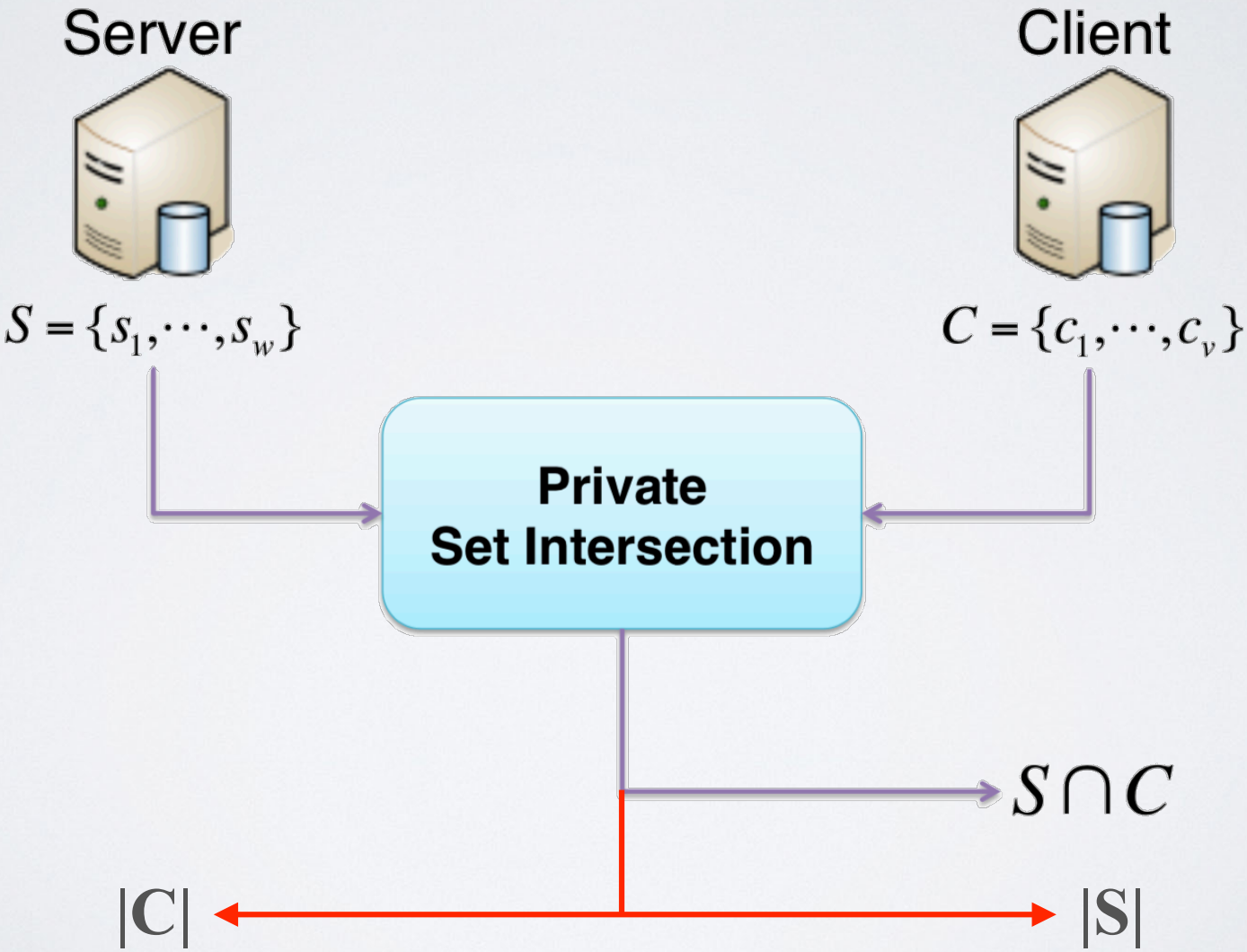
OR

Input w_1	Input w_2	Output w_3	Garbled
k_1^0	k_2^0	k_3^0	$E_{k_1^0}(E_{k_2^0}(k_3^0))$
k_1^0	k_2^1	k_3^1	$E_{k_1^0}(E_{k_2^1}(k_3^1))$
k_1^1	k_2^0	k_3^1	$E_{k_1^1}(E_{k_2^0}(k_3^1))$
k_1^1	k_2^1	k_3^1	$E_{k_1^1}(E_{k_2^1}(k_3^1))$

PRIVATE SET INTERSECTION



PRIVATE SET INTERSECTION



PERCHÉ PSI?

- Sistema di Informazione per la Sicurezza della Repubblica
- Alitalia

Sapere se ci sono terroristi su un volo

- Agenzia delle Entrate
- Banca Svizzera

Sapere se potenziali evasori hanno un conto estero

APPLICAZIONI PSI

- Test genetici
- Similarità di documenti
- Pubblicità personalizzata
- Autenticazione biometrica
- Indirizzare gli acquisti in un centro commerciale
- Individuazione di nuovi contatti in reti sociali

VARIANTI PSI

- PET (Private Equality Test),
  - Il client calcolerà 1 se e solo se S = C con |S| = |C| = 1
- PSI-CA (PSI CArdinality)
  - Il client calcolerà |S ∩ C|
- PSI-CA-T (PSI-CA Threshold)
  - Il client calcolerà 1 se e solo se |S ∩ C| > t

PROTOCOLLO SEMPLICE MA INSICURO

- Client e Server utilizzano una funzione hash h (e.g., SHA-256) per *codificare* il proprio insieme
- Il Client calcola $X_C = \{h(v) : v \in C\}$
- Il Server calcola $Y_S = \{h(v) : v \in S\}$ e lo invia al Client
- Il Client calcola $\{v \in C : h(v) \in X_C \cap Y_S\}) = C \cap S$

PROTOCOLLO SEMPLICE MA INSICURO

- Client e Server utilizzano una funzione hash h (e.g., SHA-256) per *codificare* il proprio insieme
- ~~Il Client calcola $X_C = \{h(v) : v \in C\}$~~

Il protocollo può essere semplificato, ma resta comunque insicuro

- Il Server calcola $Y_S = \{h(v) : v \in S\}$ e lo invia al Client
- Il Client calcola $\{v \in C : h(v) \in$ ~~$X_C \cap Y_S$~~$\}) = C \cap S$

PROBLEMI DI PRIVACY

- Il Client può facilmente verificare se un elemento $x \notin C$ appartiene all'insieme S
- Il Client verifica semplicemente se $h(x) \in Y_S$

PROTOCOLLO SICURO - HBC

- Basato su Blind-RSA Signatures
- Il Client fa firmare al Server, in modo **blind**, gli elementi del suo insieme
- Il Server possiede una coppia di chiavi RSA (N, e, d)

$N = p \cdot q$

- Il Client conosce la chiave pubblica (N, e)

[DCT12]

PROTOCOLLO SICURO - HBC

- Per ogni $v \in C$
  - Il Client calcola $h(v) \cdot r^e$, con r scelto a caso, e lo invia al Server
  - Il Server calcola $(h(v) \cdot r^e)^d = h(v)^d \cdot r^{e \cdot d} = h(v)^d \cdot r$ e lo invia al Client
  - Il Client conserva la coppia $(v, h(v)^d)$

$h(v)^d$ firma RSA di $h(v)$ del Client

- Il Server calcola l'insieme $Y_S = \{h(v)^d : v \in S\}$ e lo invia al Client
- Il Client calcola $\{v \in C : h(v)^d \in Y_S\}) = C \cap S$

PROTOCOLLO BASATO SU POLINOMI

- Strumenti: Schema di cifratura omomorfa a chiave pubblica
  - Semanticamente sicuro
  - Preserva l'omomorfismo dell'addizione
  - Permette moltiplicazioni per una costante

Paillier
ElGamal
Damgard&Jurik

- Insiemi rappresentati come polinomi, un elemento è uno zero del polinomio

[FNP04]

OMOMORFISMO

- Dati $Enc(k, m_1)$ ed $Enc(k, m_2)$, senza conoscere k, possiamo calcolare efficientemente
  - $Enc(k, m_1 + m_2)$
    - Ad esempio come $Enc(k, m_1) \times Enc(k, m_2)$
  - $Enc(k, c \cdot m)$
    - Ad esempio come $Enc(k, m)^c$

POLINOMIO CIFRATO

- Date le cifrature omomorfe dei coefficienti $a_0, a_1,...,a_k$ di un polinomio P di grado k
- Possiamo calcolare efficientemente una cifratura di P(y) per un qualsiasi valore in chiaro y

$$P(x) \quad = \quad a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k$$

$$P(x) \quad = \quad a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k$$

$$Enc[a_0],\ Enc[a_1],\ Enc[a_2],\ \cdots,\ Enc[a_k]$$

$$P(x) \quad = \quad a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k$$

$$Enc[a_0],\ Enc[a_1],\ Enc[a_2],\ \cdots,\ Enc[a_k]$$

Dato y calcolare $Enc[P(y)]$

$$P(x) \quad = \quad a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k$$

$$Enc[a_0],\ Enc[a_1],\ Enc[a_2],\ \cdots,\ Enc[a_k]$$

Dato y calcolare $Enc[P(y)]$

$$\begin{aligned} Enc[P(y)] &= Enc[a_0 + a_1 y + a_2 y^2 + \cdots + a_k y^k] \\ &= Enc[a_0] \cdot Enc[a_1 y] \cdot Enc[a_2 y^2] \cdots Enc[a_k y^k] \\ &= Enc[a_0] \cdot Enc[a_1]^y \cdot Enc[a_2]^{y^2} \cdots Enc[a_k]^{y^k} \end{aligned}$$

RAPPRESENTAZIONE INSIEME

- Rappresentiamo un insieme $C = \{c_1, c_2, \ldots, c_k\}$ tramite un polinomio P(x) di grado k
  - Gli elementi di C sono le radici di P(x)

- $P(x) = (x-c_1)(x-c_2)\cdots(x-c_k)$

PROTOCOLLO - CLIENT SIDE

- Il Client pubblica i parametri e la sua chiave pubblica di uno schema di cifratura omomorfo semanticamente sicuro
- Il Client possiede l'insieme $C = \{c_1, c_2, \ldots, c_k\}$
- Il Client calcola il polinomio

$$P(x) = (x - c_1) \cdot (x - c_2) \cdots (x - c_k) = \sum_{i=0}^{k} a_i x^i$$

- Il Client invia al Server i valori

  $Enc(a_0), Enc(a_1), ..., Enc(a_k)$

PROTOCOLLO - SERVER SIDE

- Il Server, per ogni $v \in S$
  - Calcola Enc[P(v)]
  - Genera un valore a caso r
  - Calcola Enc[r · P(v)+v]
    - $Enc[P(v)]^r \cdot Enc[v] = Enc[r \cdot P(v)+v]$
- Il server permuta le cifrature ottenute e le invia al Client

OSSERVAZIONE

- Se $v \in C \cap S$, allora $Enc[r \cdot P(v)+v]$ è una cifratura di v

- Se $v \notin C \cap S$ allora $Enc[r \cdot P(v)+v]$ è un valore casuale

PROTOCOLLO - CLIENT SIDE

- Il Client decifra tutti i messaggi ricevuti dal Server
- Se il Client ottiene un valore v che appartiene a C allora v apparterrà anche a $C \cap S$

NOTA

- Il Server, per ogni $v \in S$, invece di calcolare

  $$Enc[\, r \cdot P(v) + v \,]$$

  potrebbe calcolare

  $$Enc[\, r \cdot P(v) + v \,||\, payload(v) \,]$$

  dove payload(v) sono delle informazioni associate ad v

- Il client, oltre a conoscere quali $v \in C \cap S$ conoscerà anche payload(v)

PROTOCOLLO PER PSI-CA

- Il server, per ogni $v \in S$, invece di calcolare

  $$Enc[\, r \cdot P(v) + v \,]$$

  calcola

  $$Enc[\, r \cdot P(v) + 0^+ \,]$$

  0^+ è una stringa speciale che codifica il fatto che $v \in C \cap S$

  - $0+$ potrebbe consistere in una stringa con un numero prefissato di zeri
- Il Client conta il numero di occorrenze di 0^+ presenti nella della lista dei valori decifrati inviati dal Server

APPLICAZIONI PSI

- Test genetici
- Similarità di documenti
- Pubblicità personalizzata
- Autenticazione biometrica
- Indirizzare gli acquisti in un centro commerciale
- Individuazione di nuovi contatti in reti sociali

TEST DI PATERNITÀ

- Alice vuole verificare se è figlia di Bob
- Alice e Bob non vogliono scambiare le proprie sequenze di DNA
- Il DNA può essere visto come una stringa composta dai caratteri A,T,C,G
- Invece di confrontare le sequenze di DNA di Alice e Bob confrontiamo una loro codifica specifica per il test

[BBDCGT11]

CODIFICA

- Usiamo, in digitale, la stessa tecnica utilizzata per il test in vitro basata su enzimi e marcatori
- Gli enzimi spezzano le sequenza di DNA in frammenti
  - L'enzima CTGCAG se compare nella sequenza di DNA la spezza in due. Una termina con CTGCA e l'altra inizia con G
- Un marcatore seleziona uno dei frammenti

PASSI COMUNI

- Input comuni: insieme di enzimi $E = \{e_1, \ldots, e_k\}$, insieme di marcatori $M = \{m_1, \ldots, m_p\}$, una soglia τ
- Alice e Bob applicano gli enzimi in E per frammentare il DNA e usano i marcatori M per selezionare p frammenti
- Alice calcola $F_A = \{(|frag_i^A|, m_i) : 1 \leq i \leq p\}$
- Bob calcola $F_B = \{(|frag_i^B|, m_i) : 1 \leq i \leq p\}$

$frag_i^A$ è selezionato dal marcatore m_i

PROTOCOLLO

- Alice (i.e., il Client) e Bob (i.e., il Server) eseguono un protocollo PSI-CA per calcolare $|F_A \cap F_B|$
- Se $|F_A \cap F_B|$ è maggiore della soglia τ fissata per il test, allora Alice è figlia di Bob
- Non utilizziamo PSI per rilasciare meno informazioni possibili

SIMILARITÀ DI INSIEMI

- Dati due insiemi S e C, vogliamo calcolare un indice che indichi quanto i due insiemi siano simili
  - Un valore prossimo ad **uno** indica che gli insiemi sono molto **simili**
  - Un valore prossimo allo **zero** indica che gli insiemi sono molto **dissimili**
- **Applicazioni**: similarità di documenti o file multimediali, autenticazione biometrica (iris matching)

[BDCG14]

INDICE DI JACCARD

- Utilizzato per verificare la similarità di due insiemi
- **J**(C, S) = |C ∩ S| / |C ∪ S|

  = |C ∩ S| / (|C| + |S| - |C ∩ S|)
- Ovviamente potremmo usare PSI o PSI-CA

PROBLEMI DI PRIVACY

Vogliamo calcolare solo un numero compreso tra zero e uno

- Protocollo basato su PSI
  - Rilascia informazioni su $|C|$, $|S|$ e sugli elementi in $C \cap S$
- Protocollo basato su PSI-CA
  - Rilascia informazioni su $|C|$, $|S|$ e $|C \cap S|$

UTILIZZIAMO MIN-HASH

- h_i è una funzione hash, $1 \leq i \leq k$
  - $\kappa_i = \min \{h_i(c_i) : 1 \leq i \leq v\}$
  - $\sigma_i = \min \{h_i(s_i) : 1 \leq i \leq w\}$
- $SK_C = \{(\kappa_1, 1), (\kappa_2, 2), \ ..., (\kappa_k, k)\}$ è lo ***sketch*** di C
- $SK_S = \{(\sigma_1, 1), (\sigma_2, 2), \ ..., (\sigma_k, k)\}$ è lo ***sketch*** di S

UTILIZZIAMO MIN-HASH

- h_i è una funzione hash, $1 \leq i \leq k$
  - $\kappa_i = \min \{h_i(c_i) : l \leq i \leq v\}$
  - $\sigma_i = \min \{h_i(s_i) : l \leq i \leq w\}$

$J(C, S) \cong J(SK_C, SK_S)$

errore $O(k^{-1/2})$

- $SK_C = \{(\kappa_1, 1), (\kappa_2, 2), \ ..., (\kappa_k, k)\}$ è lo ***sketch*** di C
- $SK_S = \{(\sigma_1, 1), (\sigma_2, 2), \ ..., (\sigma_k, k)\}$ è lo ***sketch*** di S

PROTOCOLLO

- Il Client e il Server applicano PSI-CA su SK_C e SK_C
- Il Client apprende $|SK_C \cap SK_C|$ e calcola $J(C, S)$ come $|SK_C \cap SK_C|/k$
- Non c'è perdita di informazione come nel protocollo precedente
  - Si calcola un'approssimazione dell'indice di Jaccard

PSI-CA APPROSSIMATO

- Possiamo utilizzare il protocollo precedente per calcolare un'approssimazione di $|C \cap S|$ purché il Client conosca $|S|$
- Con il protocollo precedente il Client apprende $\delta = |SK_C \cap SK_C|$
- Il Server invia $w = |S|$ al Client che conosce $v = |C|$
- Il Client calcola $\delta \cdot (v + w)/(1 + \delta) \cong |C \cap S|$

SIMILARITÀ DI DOCUMENTI

- Rappresentiamo il documento con i suoi trigrammi (sequenze di tre caratteri consecutivi)

SIMILARITÀ DI DOCUMENTI

- Rappresentiamo il documento con i suoi trigrammi (sequenze di tre caratteri consecutivi)

the quick brown fox jumps over the lazy dog

SIMILARITÀ DI DOCUMENTI

- Rappresentiamo il documento con i suoi trigrammi (sequenze di tre caratteri consecutivi)

azy, bro, ckb, dog, ela, equ, ert, fox, hel, heq, ick, jum, kbr, laz, mps, nfo, ove, own, oxj, pso, qui, row, rth, sov, the, uic, ump, ver, wnf, xju, ydo, zyd

SIMILARITÀ DI DOCUMENTI

- Rappresentiamo il documento con i suoi trigrammi (sequenze di tre caratteri consecutivi)

azy, bro, ckb, dog, ela, equ, ert, fox, hel, heq, ick, jum, kbr, laz, mps, nfo, ove, own, oxj, pso, qui, row, rth, sov, the, uic, ump, ver, wnf, xju, ydo, zyd

- Utilizziamo l'indice di Jaccard

PUBBLICITÀ PERSONALIZZATA

- Attori
  - **Advertiser** che ha una pubblicità
  - **Online Social Platform** che vuole mostrare la pubblicità ad utenti selezionati
- Pubblicità e utente rappresentati con un elenco di argomenti

PUBBLICITÀ PERSONALIZZATA

- Attori
  - **Advertiser** che ha una pubblicità
  - **Online Social Platform** che vuole mostrare la pubblicità ad utenti selezionati
- Pubblicità e utente rappresentati con un elenco di argomenti

tweet

The Role of Artificial Intelligence in Growing Business #Machine-learning #Deep-Learning #AI #ML …

PUBBLICITÀ PERSONALIZZATA

- Attori
  - **Advertiser** che ha una pubblicità
  - **Online Social Platform** che vuole mostrare la pubblicità ad utenti selezionati
- Pubblicità e utente rappresentati con un elenco di argomenti

tweet

The Role of Artificial Intelligence in Growing Business #Machine-learning #Deep-Learning #AI #ML …

PUBBLICITÀ PERSONALIZZATA

- Attori
  - **Advertiser** che ha una pubblicità
  - **Online Social Platform** che vuole mostrare la pubblicità ad utenti selezionati
- Pubblicità e utente rappresentati con un elenco di argomenti

tweet

The Role of Artificial Intelligence in Growing Business #Machine-learning #Deep-Learning #AI #ML …

argomenti

Artificial Intelligence, Business, …

INPUT CLIENT E SERVER

- **Advertiser** (Client) $A = \{a_1, a_2, \ldots, a_k\}$
- **OSP** (Server) $U = \{u_1, u_2, \ldots, u_t\}$
  - **OSP** calcola $R_i = \{u \in U : \mathbf{sr}(u_i, u) \geq \varepsilon\}$, sottoinsiemi di argomenti che sono correlati secondo la misura **sr** (*semantic relatedness*)

MOSTRARE LA PUBBLICITÀ?

- **OSP** calcola gli insiemi $L = |\{u_i \in U : R_i \subseteq A\}|$ e $E = |\{u_i \in U : R_i \cap A = \emptyset\}|$. Se $2 \cdot \frac{L^2 + L \cdot E}{t(k+t)} > \lambda$ si mostra la pubblicità all'utente

MOSTRARE LA PUBBLICITÀ?

- **OSP** calcola gli insiemi $L = |\{u_i \in U : R_i \subseteq A\}|$ e $E = |\{u_i \in U : R_i \cap A = \varnothing\}|$. Se $2 \cdot \frac{L^2 + L \cdot E}{t(k+t)} > \lambda$ si mostra la pubblicità all'utente

- $R_i \subseteq A \iff |R_i \cap A| = |R_i|$

  $R_i \cap A = \varnothing \iff |R_i \cap A| = 0$

MOSTRARE LA PUBBLICITÀ?

- **OSP** calcola gli insiemi $L = |\{u_i \in U : R_i \subseteq A\}|$ e $E = |\{u_i \in U : R_i \cap A = \varnothing\}|$. Se $2 \cdot \frac{L^2 + L \cdot E}{t(k+t)} > \lambda$ si mostra la pubblicità all'utente

- $R_i \subseteq A \iff |R_i \cap A| = |R_i|$

  $R_i \cap A = \varnothing \iff |R_i \cap A| = 0$

PSI-CA

PROTOCOLLO PRIVATO

Advertiser (Client)

$A = \{a_1, a_2, \ldots, a_k\}$

OSP (Server)

$\{R_1, R_2, \ldots, R_t\}$

PROTOCOLLO PRIVATO

Advertiser (Client)

$A = \{a_1, a_2, \ldots, a_k\}$

OSP (Server)

$\{R_1, R_2, \ldots, R_t\}$

Per $i = 1, \ldots, t$

PROTOCOLLO PRIVATO

Advertiser (Client)

$A = \{a_1, a_2, \ldots, a_k\}$

OSP (Server)

$\{R_1, R_2, \ldots, R_t\}$

Per $i = 1, \ldots, t$

$d_i = |R_i \cap A|, |R_i|$ ⟷ PSI-CA(A, R_i) ⟷ $|A|$

PROTOCOLLO PRIVATO

Advertiser (Client)

$A = \{a_1, a_2, \ldots, a_k\}$

OSP (Server)

$\{R_1, R_2, \ldots, R_t\}$

Per $i = 1, \ldots, t$

PSI-CA(A, R_i)

$d_i = |R_i \cap A|, |R_i|$ ⟷ $|A|$

Se $d_i = |R_i|$, allora L=L+1

Se $d_i = 0$, allora E=E+1

PROTOCOLLO PRIVATO

Advertiser (Client)

$A = \{a_1, a_2, \ldots, a_k\}$

OSP (Server)

$\{R_1, R_2, \ldots, R_t\}$

Per $i = 1, \ldots, t$

PSI-CA(A, R_i)

$d_i = |R_i \cap A|, |R_i|$ ⟷ $|A|$

Se $d_i = |R_i|$, allora L=L+1

Se $d_i = 0$, allora E=E+1

$$2 \cdot \frac{L^2 + L \cdot E}{t(k+t)}$$ ⟶

OSSERVAZIONI

- Per problemi di privacy non possiamo invertire il ruolo dell'**Advertiser** e dell'**OSP**
- Il protocollo rilascia all'**Advertiser** delle informazioni aggiuntive, la dimensione di R_i
  - Risolto con l'utilizzo di una variante dello schema di cifratura di El Gamal
  - Esiste anche una versione *outsourced* del protocollo

GRAZIE PER L'ATTENZIONE

(POCHI) RIFERIMENTI

C. Blundo, E. De Cristofaro, and P. Gasti. EsPRESSO: Efficient privacy-preserving evaluation of sample set similarity. *Journal of Computer Security*, 22(3):355–381, 2014.

C. Blundo, C. D. Maio, M. Parente, and L. Siniscalchi. An intelligent and private method to profile social network users. In *IEEE International Conference on Fuzzy Systems*, 2019.

C. Blundo, F. Orciuoli, and M. Parente. An ami-based and privacy-preserving shopping mall model. *Human-centric Computing and Information Sciences*, 7:26, 2017.

H. Chen, K. Laine, and P. Rindal. Fast private set intersection from homomorphic encryption. In *Conference on Computer and Communications Security*, pages 1243–1255, 2017.

E. De Cristofaro and G. Tsudik. Practical private set intersection protocols with linear complexity. In *Financial Cryptography and Data Security*, pages 143–159, 2010.

E. De Cristofaro and G. Tsudik. Experimenting with fast private set intersection. In *Trust and Trustworthy Computing*, pages 55–73, 2012.

D. Demmler, P. Rindal, M. Rosulek, and N. Trieu. PIR-PSI: scaling private contact discovery. *PoPETs*, 2018(4):159–178, 2018.

M. Fischlin, B. Pinkas, A. Sadeghi, T. Schneider, and I. Visconti. Secure set intersection with untrusted hardware tokens. In *Topics in Cryptology - CT-RSA*, pages 1–16, 2011.

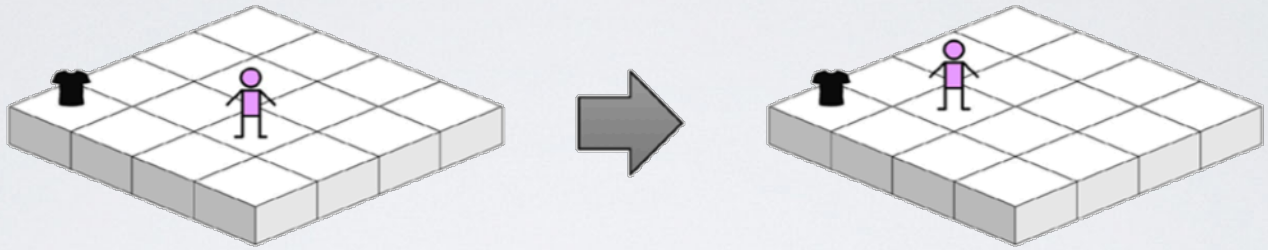
M. J. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *Advances in Cryptology - EUROCRYPT*, pages 1–19, 2004.

S. Kamara, P. Mohassel, M. Raykova, and S. S. Sadeghian. Scaling private set intersection to billion-element sets. In *Financial Cryptography and Data Security*, pages 195–215, 2014.

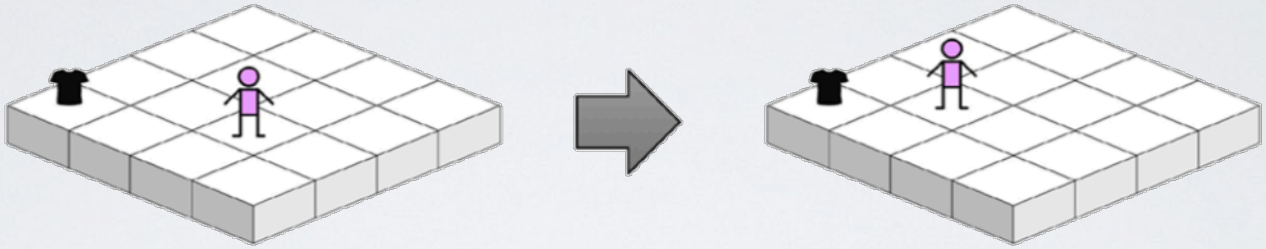
B. Pinkas, M. Rosulek, N. Trieu, and A. Yanai. Spot-light: Lightweight private set intersection from sparse OT extension. In *Advances in Cryptology - CRYPTO*, pages 401–431, 2019.

B. Pinkas, T. Schneider, and M. Zohner. Faster private set intersection based on OT extension. In *USENIX Security Symposium*, pages 797–812, 2014.

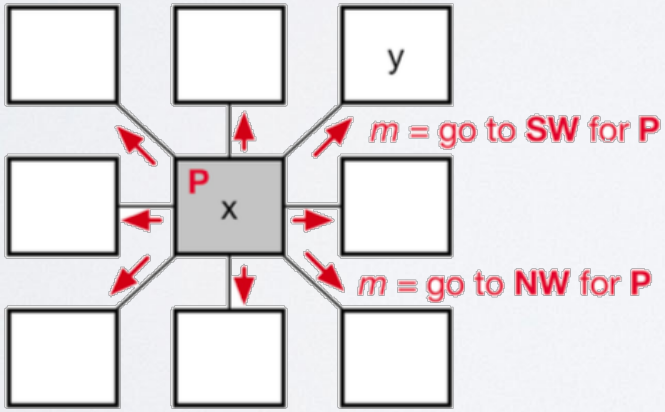
PRIVACY PRESERVING SHOPPING MALL



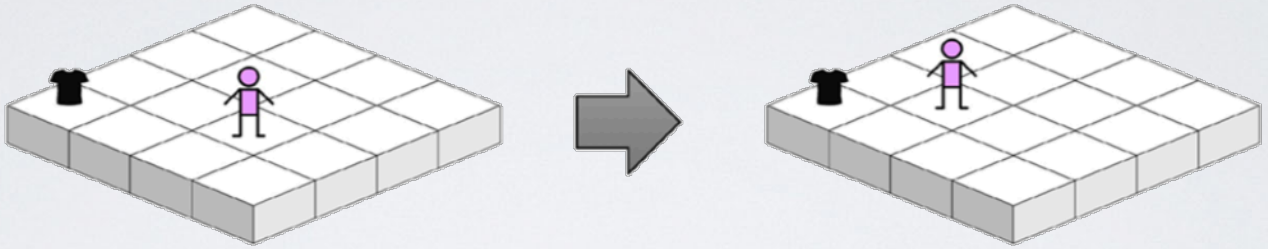
PRIVACY PRESERVING SHOPPING MALL



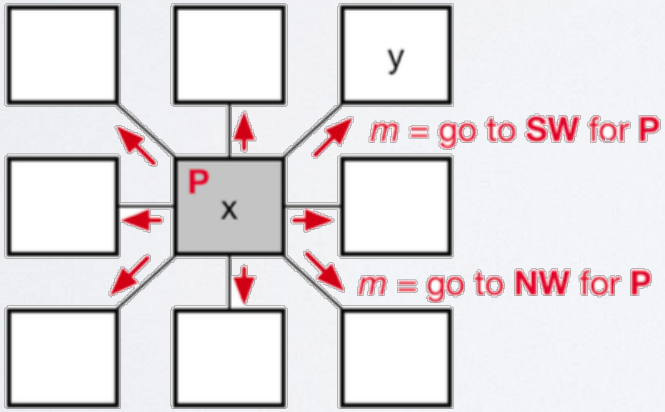
I-HAVE-PRODUCT



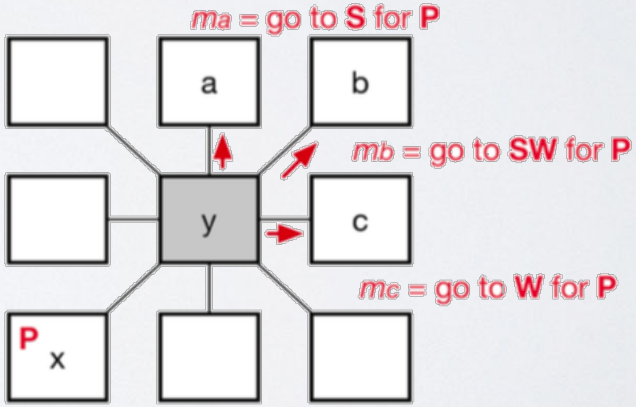
PRIVACY PRESERVING SHOPPING MALL



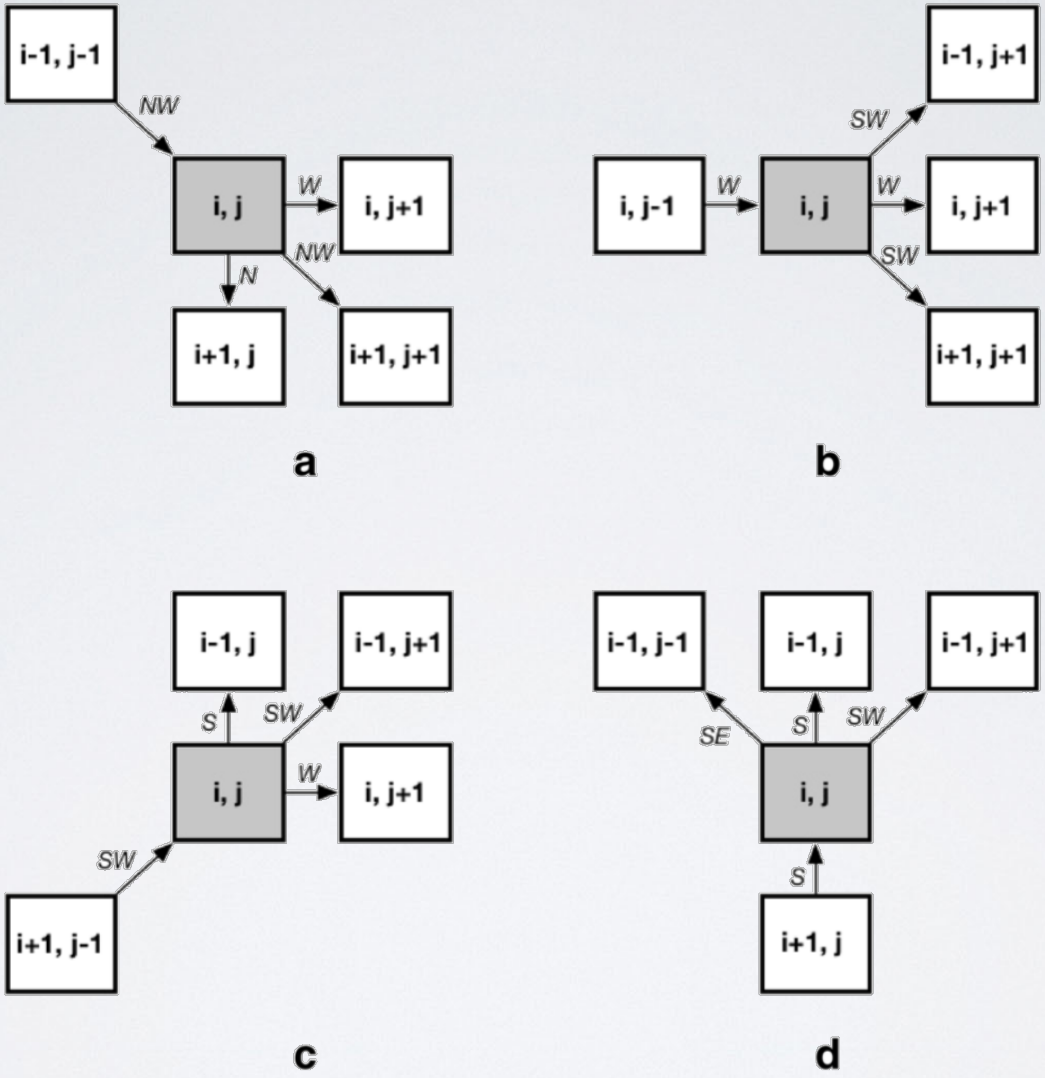
I-HAVE-PRODUCT



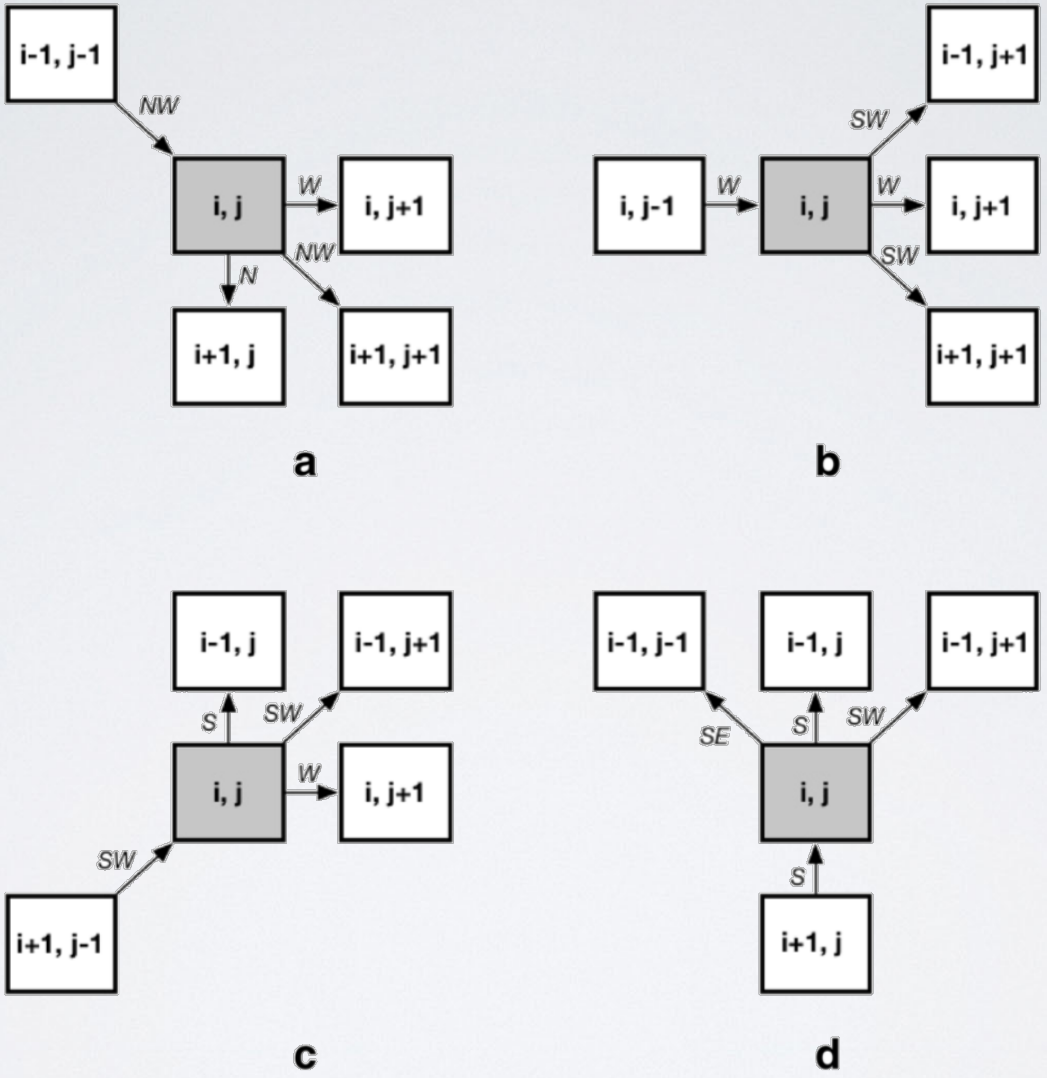
RELAY



PROPAGAZIONE MESSAGGI



PROPAGAZIONE MESSAGGI



$msg_{NE} = prod_{SHOP} \cup msg_{W} \cup msg_{SW} \cup msg_{S}$

INDIVIDUAZIONE PRODOTTI

- u_{SL} = lista della spesa dell'utente u
- $u_{SL} \cap prod_{SHOP}$ = prodotti che sta cercando l'utente e che sono disponibili nel negozio **SHOP**
- $u_{SL} \cap prod_{D}$ = prodotti che sta cercando l'utente e che sono disponibili in direzione **D**

CIFRATURA COMMUTATIVA DETERMINISTICA SIMMETRICA

- **Init**$(1^\lambda) \rightarrow$ pp
- **KeyGen**(pp) $\rightarrow$ sk
- **Enc**(sk, m) $\rightarrow$ c
- **Dec**(sk, c) $\rightarrow$ m'

CIFRATURA COMMUTATIVA DETERMINISTICA SIMMETRICA

- **Init**$(1^\lambda) \rightarrow$ pp
- **KeyGen**(pp) $\rightarrow$ sk
- **Enc**(sk, m) $\rightarrow$ c
- **Dec**(sk, c) $\rightarrow$ m'

Dec(sk, **Enc**(sk, m)) = m

CIFRATURA COMMUTATIVA DETERMINISTICA SIMMETRICA

- **Init**$(1^\lambda) \rightarrow$ pp
- **KeyGen**(pp) $\rightarrow$ sk
- **Enc**(sk, m) $\rightarrow$ c
- **Dec**(sk, c) $\rightarrow$ m'

Dec(sk, **Enc**(sk, m)) = m

Enc(sk, **Enc**(sk', m))

=

Enc(sk', **Enc**(sk, m))

INIZIALIZZAZIONE

- Tutti i negozi comunicano al gestore del centro commerciale la lista dei prodotti in offerta con il relativo prezzo
- Il gestore del centro commerciale seleziona, per ogni tipologia di prodotto, quello di prezzo minimo e invia al negozio che lo vende **ENC**(sk, **id**)
  - L'insieme dei valori cifrati costituisce $prod_{SHOP}$

UTENTE NEL CENTRO COMMERCIALE

- L'utente invia al gestore del centro commerciale la lista {**ENC**(sk', **id**) : **id** ∈ lista della spesa}
- Il gestore del centro commerciale invia all'utente la lista {**ENC**(sk, **ENC**(sk', **id**))}
- L'utente, dall lista ricevuta, deriva u_{SL} = {**ENC**(sk, **id**)}

NEGOZIO ⇄ UTENTE

- Il negozio (Server) conosce $prod_{SHOP}$, msg_{N}, msg_{NE}, msg_{E}, msg_{SE}, msg_{S}, msg_{SW}, msg_{W}, msg_{NW}
- L'utente (Client) conosce u_{SL}
- Il negozio e l'utente interagiscono nove volte tramite un protocollo di PSI o PSI-CA
  - L'esito del protocollo determina l'acquisto di prodotti nel negozio e/o lo spostamento verso altre direzioni