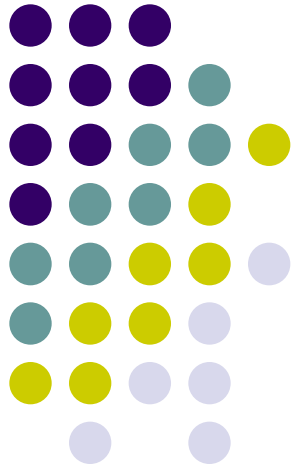
Secure Multi-Party Computation

(cryptography for the not so good, the not so bad and the not so ugly)

María Isabel González Vasco
mariaisabel.vasco@urjc.es

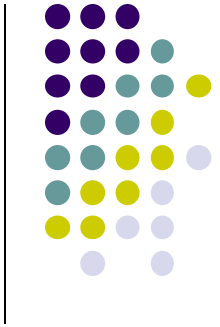


Based on joint work with **Paolo D'Arco** (U. Salerno) **Angel L. Pérez del Pozo** (URJC) and **Claudio Soriente** (ETH Zürich)

Universidad Rey Juan Carlos

Outline

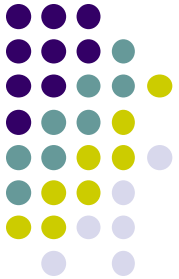
- Secure Multiparty Computation
  - Definitions
  - Security Models and Generic results
  - Example: secure addition
- Private Set intersection :
  - Unconditional
  - Computational
    - OPE
    - OPRF
    - Our work in size-hiding



SECURE MULTIPARTY COMPUTATION

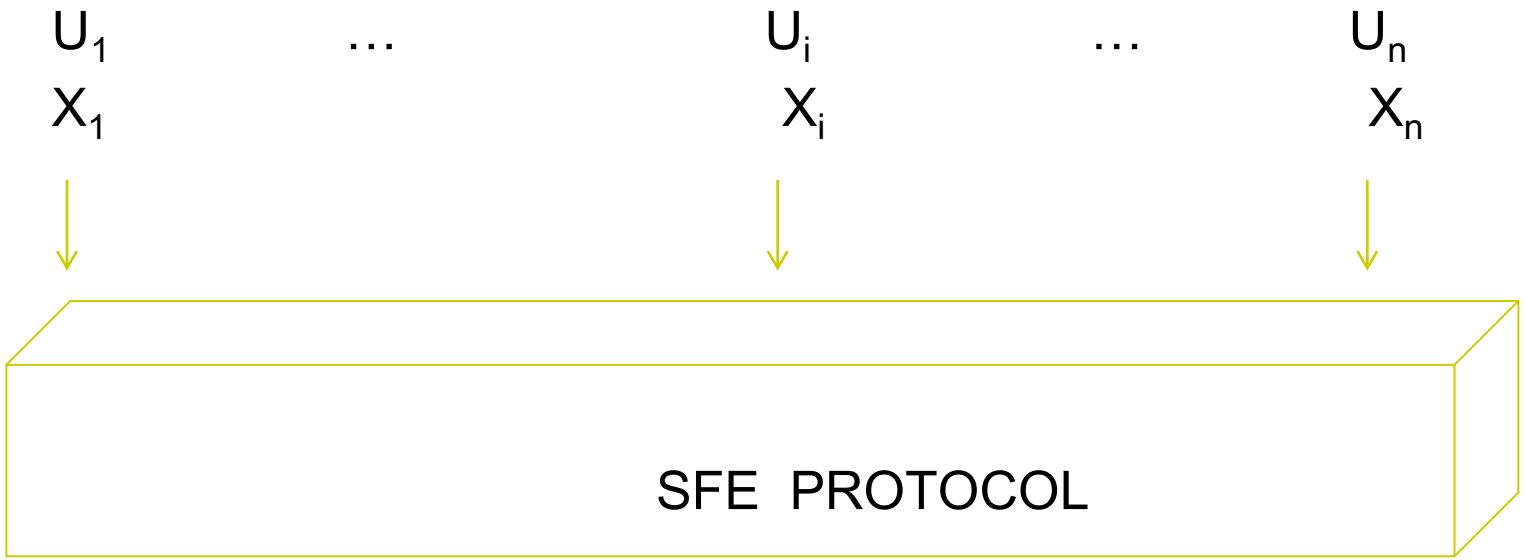
M.I. Glez. Vasco
Marzo 2015

What is Secure Multi-Party Computation?

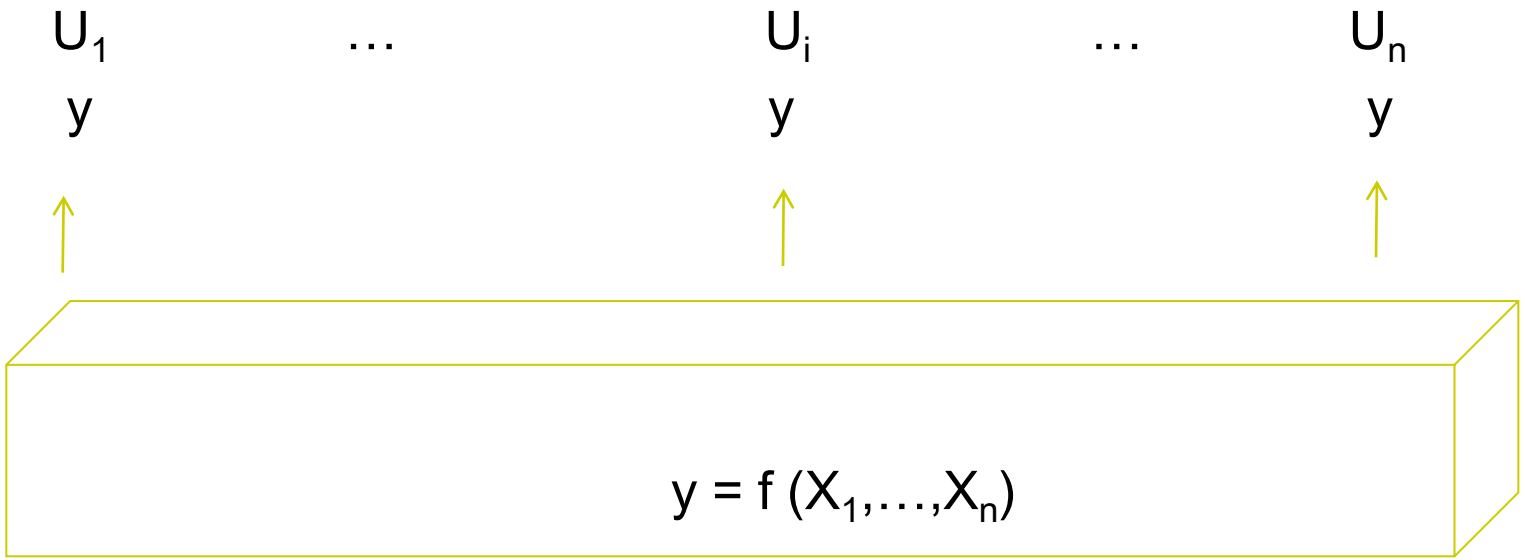


- U_1 ,…, U_n a set of participants aiming at a common goal…but with conflicting interests
- Users want to attain correctness and ‘certain’ security guarantees
- Security must be preserved in the face of adversarial behavior by some of the participants, or by an external party (or both!!) .

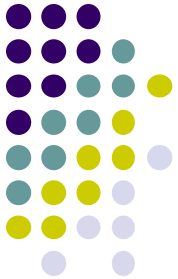
Secure MPC



Secure MPC

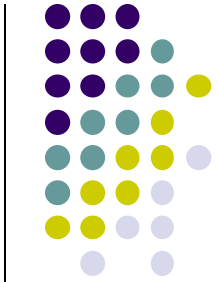


Aplication scenarios



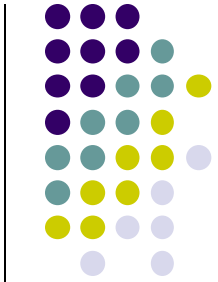
- ELLECTIONS: ellectors want to be assured that their private input (vote) will be counted, yet, they should not be able to prove to a third party which their vote was (coercion).
- PROCUREMENT: ``inverted auction'', public institution asks companies to bid for a contract; they would rather not reveal the minimum paymment they would accept to do the job.
- AUCTIONS: all bidders want to keep private the maximum value they are willing to pay for the item on sale (yet, the result of the aution may in principle be computed from this private input).
- DATA MINING/PIR: Coordinated access to certain sensitive databases (from tax authorities, healthcare system)

The role of cryptography



- Finding efficient designs in the right security model:
  - A secure protocol must reveal no more information than the output of the function itself (the process of protocol computation reveals nothing).
  - Components of ANY security definition
    - Adversarial power
    - Network model
      - Type of network
      - Existence of trusted help
      - Stand-alone versus composition
    - Security guarantees aimed at
- Deciding which functions to compute is a different challenge...

Security Models



- Information Theoretic (Unconditional) setting:

  no computational assumption.

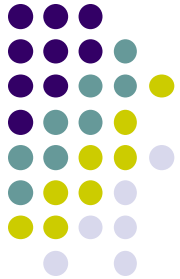
- Computational (Cryptographic) setting:

  we add computational assumptions, consider pptm participants, and may also:

  - Rely on harndness assumptions related to number theoretical problems
  - Assume the existence of ideal hash functions (random oracles)

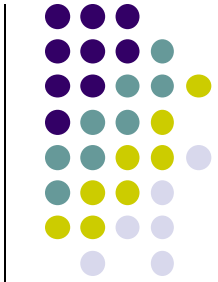
- Further relaxations (of both models)
  - Assume the presence of a Trusted Third Party

Generic results: Computational Setting



- Any two-party function can be securely computed in the semi-honest adversarial model Yao, [86]
- Any multiparty function can be securely computed in the malicious model, for an honest majority Goldwasser et al. [87]
- Remarks:
  - Above results assume the existence of trapdoor permutations
  - The above can be achieved in constant-round, Beaver et al. [90]
  - With an honest majority, fairness and guaranteed output delivery are achieved. Otherwise, not.
  - Model: static corruptions, authenticated channels, polynomial-time adversary, stand-alone…

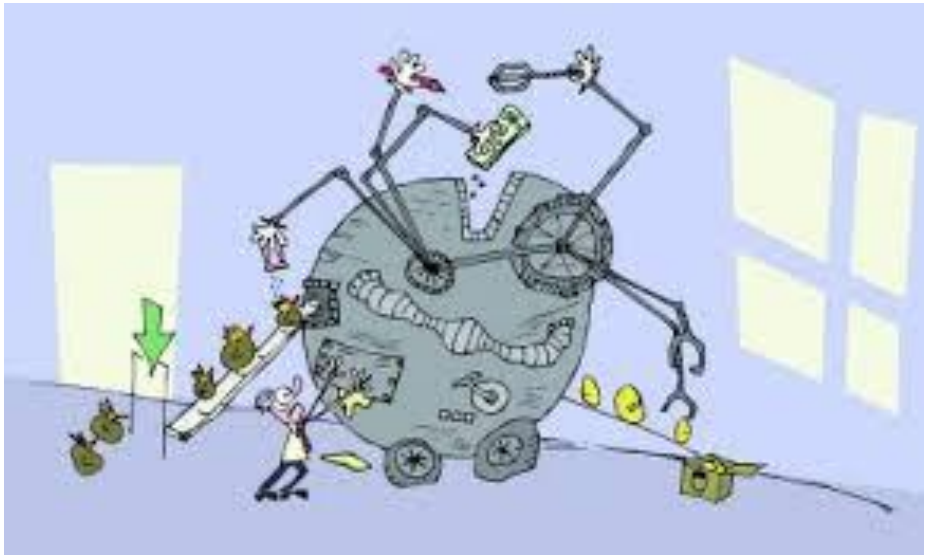
Generic results: Information Theoretic Setting



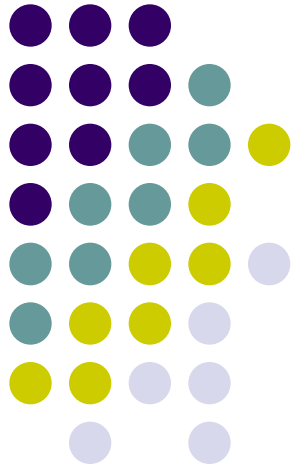
- Assuming a 2/3 honest majority, any multiparty function can be securely computed in the malicious model, Ben-Or et al., Chaum et al. [88]
- Assuming a (regular) honest majority and a broadcast channel, any multiparty function can be securely computed in the malicious model Rabin et al. [89]
- Unconditionally secure MPC follows from 'special' secret sharing schemes. Cramer et al. [2000]
- Remarks:
  - Above results do not assume any complexity assumptions
  - Model: adaptive corruptions, perfectly private and authenticated channels, unbounded adversary, stand-alone…

Interpretation

- In the models described above, any distributed task can be securely computed.
- Finding an efficient solution is however another issue…

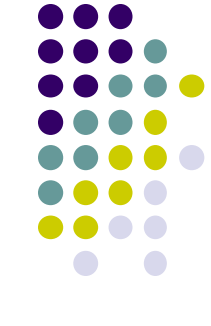


A simple example:
Secure addition



M.I. Glez. Vasco
Marzo 2015

Basic tool: SSS



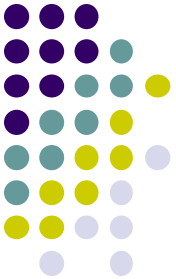
Secret Sharing Schemes, Shamir, Blackley [79]

allow a party U_1 to spread information on a secret value s across all participants

U_2 ,…, U_n

s.t., all together hold full info on s (yet individuals have no info at all)

Sharing a value $s \in \mathbb{Z}_p$



- U_1 chooses $r_1, \ldots, r_{n-1}$ uniformly at random in $\mathbb{Z}_p$, and sets

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

- U_1 sends privately to each U_j, all r_i for all $i \neq j$
- Clearly, any coalition of two participants may recover s
- On the other hand, each U_j, for $1 \neq j$, has no information on the secret s

Secure Addition

- Each U_i distributes shares of his secret s_i as in the previous protocol, namely,
  - U_i chooses $r_{i,1}$,…, $r_{i,n-1}$ uniformly at random in $\mathbb{Z}_p$, and sets

    $r_{i,n} = s_i - r_{i,1} - \ldots - r_{i,n-1}$
  - U_i sends privately to U_j, all $r_{i,k}$ for all $k \neq j$
- Each U_i adds corresponding shares of each secret, namely, for each $k \neq i$ he computes $sum_k = r_{1,k} + r_{2,k} + \ldots + r_{n,k}$

All parties compute the result as

$sum = sum_1 + \ldots + sum_n$

Secure addition (3-party)

$r_{1,3}$, $r_{1,1}$

$r_{2,3}$, $r_{2,2}$

$r_{1,3} = s1 - r_{1,1} - r_{1,2}$

$r_{1,1}$, $r_{1,2}$

$r_{3,2}$ $r_{3,3}$

$r_{3,1}$, $r_{3,3}$

$r_{2,3} = s2 - r_{2,1} - r_{2,2}$

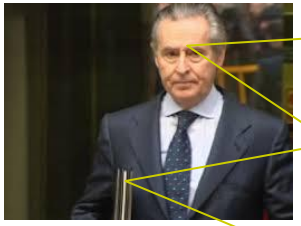
$r_{2,1}$, $r_{2,2}$

$r_{3,3} = s3 - r_{3,1} - r_{3,2}$

Secure addition (3-party)

$sum_2 = r_{1,2} + r_{2,2} + r_{3,2}$

$sum_3 = r_{1,3} + r_{2,3} + r_{3,3}$



s1 +s2 + s3 =

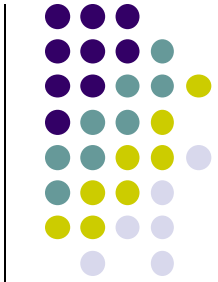
$sum_1 + sum_2 + sum_3$

$sum_1 = r_{1,1} + r_{2,1} + r_{3,1}$

$sum_3 = r_{1,3} + r_{2,3} + r_{3,3}$

$sum_2 = r_{1,2} + r_{2,2} + r_{3,2}$

$sum_1 = r_{1,1} + r_{2,1} + r_{3,1}$



PRIVATE SET INTERSECTION

Based on joint work with **Paolo D'Arco** (U. Salerno) **Angel L. Pérez del Pozo** (URJC) and **Claudio Soriente** (ETH Zürich)

M.I. Glez. Vasco
Marzo 2015

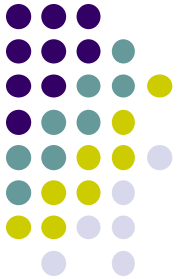
The PSI Problem: motivation

Situation: two mutually distrusting entities hold non-disjoint sets of private data and want to take some actions on the intersection.

European banks
I do not trust

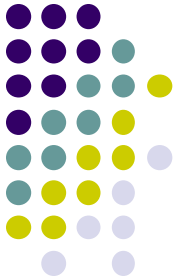
European banks
I do not trust

The PSI Problem: motivation



- Two national law enforcement bodies (FBI and MI5) want to compare their respective databases of terrorist suspects.
- A tax authority wants to learn whether any suspected tax evaders have accounts with a certain foreign bank.
- Two real estate companies are willing to identify customers who are double dealing.
- An on-line dating system allows you to search for people with shared hobbies.

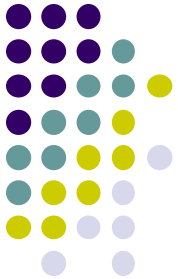
A PSI Protocol: formulation



- Players: Client (C) and Server (S), each holding a private set $S = \{s_1, \ldots, s_w\}$ and $C = \{c_1, \ldots, c_v\}$ from a ground set U. We consider them to be HBO (honest-but-curious).

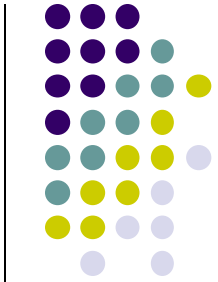
- Algorithms:
  - Setup: selects all global parameters
  - $\mathrm{Interaction}$: protocol played by C and S, on (private) input their respective sets S and C, provides private output to both parties. Typically, client gets $S \cap C$.

A PSI Protocol: formulation



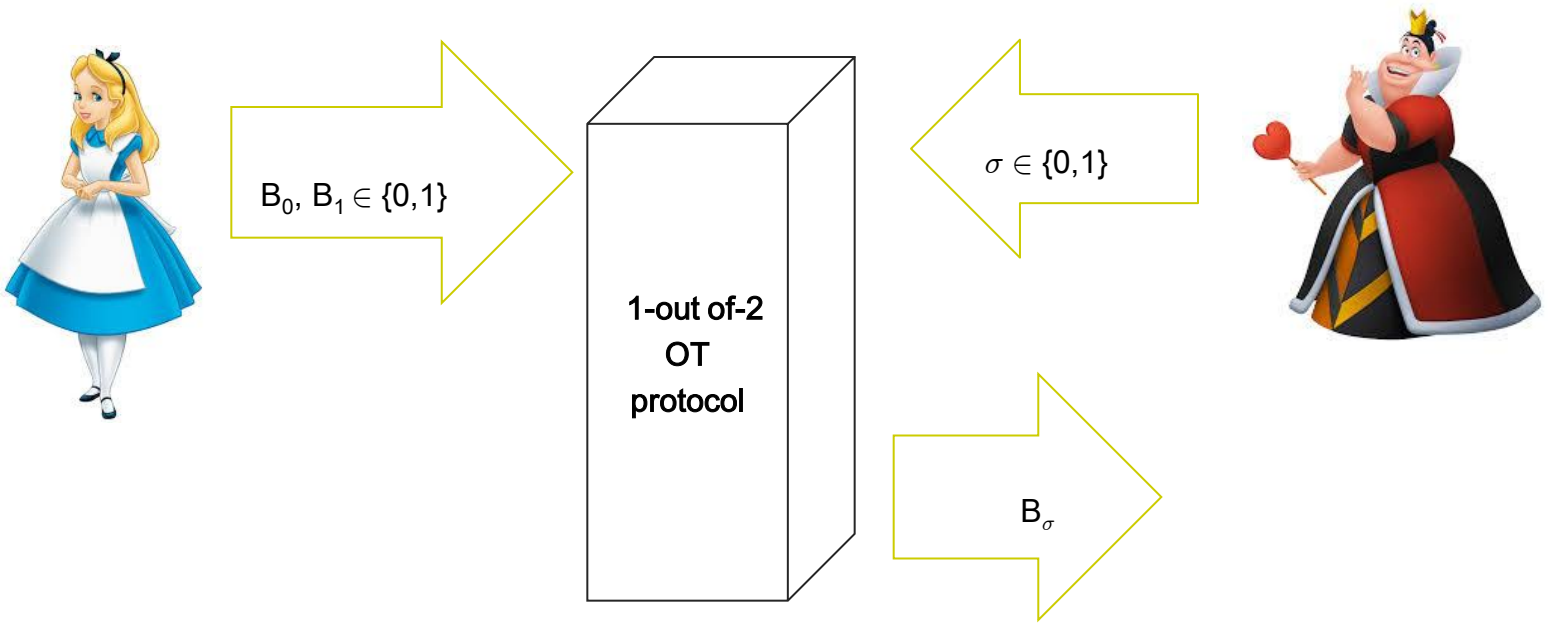
- Requirements (informal –- one-sided):
  - Correctness: once $\mathrm{Interaction}$ is executed, with overwhelming probability, C gets as private output the intersection $C \cap S$ and (maybe) $|S|$ (w).
  - Server Privacy: the execution of $\mathrm{Interaction}$ leaks no information about $S \cap C^{\mathrm{c}}$ (apart from its size)
  - Client Privacy: no information about C, apart from its size v, is leaked from $\mathrm{Interaction}$.

Unconditional setting



- PSI is impossible in the plain model, as it is equivalent to OT (Oblivious Transfer)
- Two side size hidding PSI, with a TTP, is possible in an unconditional setting

What is OT?

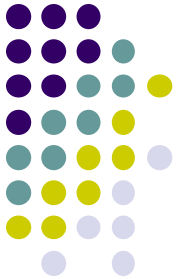


PSI versus OT

- Unconditionally secure OT is impossible
- PSI implies OT:
  - Sender generates a set as $\{0|\ b_0\ ,1|b_1\}$
  - Chooser generates a set as $\{\sigma|0,\ \sigma|1\}$
  - After running a PSI protocol, Chooser acting as 'Queen' gets $\{\sigma\ |\ b_\sigma\}$

Unconditionally secure PSI is impossible

Unconditional setting



- PSI is impossible in the plain model, as it is equivalent to Oblivious Transfer
- Two side size hiding PSI, with a TTP, is possible in an unconditional setting

Unconditionally Secure SH-PSI with a TTP

Setup: the TTP chooses two random bijections f, g: $\wp(U) \mapsto \{0,1\}^{|U|}$

Unconditionally Secure SH-PSI with a TTP

Setup: the TTP chooses two random bijections $\mathrm{f}, \mathrm{g}: \wp(U) \mapsto \{0,1\}^{|U|}$

$C = \{\mathrm{c}_1, \ldots, \mathrm{c}_v\}$

R, L

$\mathrm{R} = \mathrm{f}(C), \mathrm{L} = \{(\mathrm{g}(\mathrm{D}), \mathrm{D}) : \mathrm{D} \subseteq C\}$

Unconditionally Secure SH-PSI with a TTP

Setup: the TTP chooses two random bijections f,g: $\wp(U) \mapsto \{0,1\}^{|U|}$

$C = \{c_1, \ldots, c_v\}$

R, L

$R = f(C), L = \{(g(D), D) : D \subseteq C\}$

$T = \{(f(E), g(E \cap S)) : E \subseteq U\}$

$S = \{s_1, \ldots, s_w\}$

T

Unconditionally Secure SH-PSI with a TTP

Setup: the TTP chooses two random bijections f, g: $\wp(U) \mapsto \{0,1\}^{|U|}$

$C = \{c_1, \ldots, c_v\}$

R, L

$R = f(C), L = \{(g(D), D) : D \subseteq C\}$

$T = \{(f(E), g(E \cap S)) : E \subseteq U\}$

$S = \{s_1, \ldots, s_w\}$

T

Interaction:

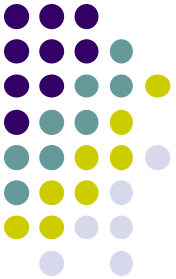
R

R'

Search for $(R, *)$ in T,
define R' as *

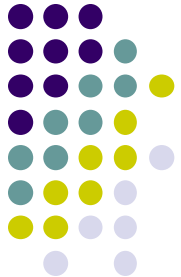
Search for (R', D) in L
Output D

Unconditionally Secure SH-PSI with a TTP



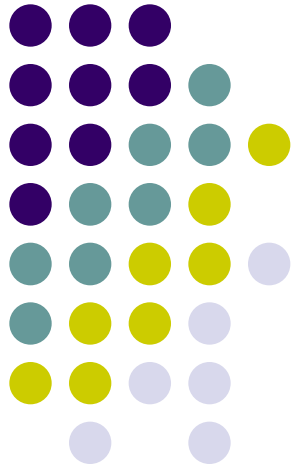
- We get:
  - Correctness: once $\mathrm{Interaction}$ is executed, C gets as private output the intersection $C \cap S$ √
  - Server Privacy: the execution of $\mathrm{Interaction}$ leaks no information about $C^c \cap$ S √ - as g is a random bijection, R' leaks nothing
  - Client Privacy: no information about C, is leaked from $\mathrm{Interaction}$. √ - as f is a random bijection, R leaks nothing

Cryptographic setting



- Technique I: Oblivious Polynomial Evaluation
  Freedman et al. [04]
- Technique II: Oblivious Pseudorandom Function Evaluation
  Hazai et al., Jarecki at al[08,10]
- What do we do if size matters?
  Ateniese et al. [11], D'Arco et al [11,..]

Technique I:
Oblivious Polynomial Evaluation



M.I. Glez. Vasco

A seminal work: Freedman et al [2004]

Setup: run by C, selects an encoding of U as a subset of Z_n^* ,

runs a key generation algorithm for Paillier encryption which outputs (s_k, p_k)

$p_{k,}$ Encoding

A seminal work: Freedman et al [2004]

Setup: run by C, selects an encoding of U as a subset of Z_n^* ,
runs a key generation algorithm for Paillier encryption which outputs (s_k, p_k)

p_k, Encoding

Interaction:

$C = \{c_1, \ldots, c_v\}$, s_k

$S = \{s_1, \ldots, s_w\}$,
$\pi \in_R S_w$

$P(x) = \Pi_1^v (x - c_i) = \Sigma_0^k \alpha_i t^i$ $\quad \{ Enc(\alpha_i) \}_{i=0}^k$ →

A seminal work: Freedman et al [2004]

Setup: run by C, selects an encoding of U as a subset of Z_n^* ,
runs a key generation algorithm for Paillier encryption which outputs (s_k, p_k)

p_k, Encoding

Interaction:

$C = \{c_1, \ldots, c_v\}$, s_k

$S = \{s_1, \ldots, s_w\}$,
$\pi \in_R S_w$

$P(x) = \Pi_1^v (x - c_i) = \Sigma_0^k \alpha_i t^i$

$\{ Enc(\alpha_i) \}_{i=0}^k$ →

for i=1 to w
$r_i \in_R Z_n^*$
$e_i = Enc(r_i P(s_i) + s_i)$

← $\{ \pi(e_i) \}_{i=1}^w$

A seminal work: Freedman et al [2004]

Setup: run by C, selects an encoding of U as a subset of Z_n^* ,
runs a key generation algorithm for Paillier encryption which outputs (s_k, p_k)

p_k, Encoding

Interaction:

$C = \{c_1,\ldots, c_v\}$, s_k

$S = \{s_1,\ldots, s_w\}$
$\pi \in_R S_w$

$P(x) = \Pi_1^v (x - c_i) = \Sigma_0^k \alpha_i t^i$

$\{ Enc(\alpha_i) \}_{i=0}^k$ →

for i=1 to w
$r_i \in_R Z_n^*$
$e_i = Enc(r_i P(s_i) + s_i)$

← $\{ \pi(e_i) \}_{i=1}^w$

$C \cap S := \{c_1,\ldots, c_v\} \cap \{ Dec(e_i) \}_{i=1}^w$

Underlying encryption

- OPE calls for an additively homomorphic encryption scheme, so that $\mathrm{Enc}(r_i\, P(s_i) + s_i)$ may be computed from the encryptions of P's coefficients
- Paillier encryption --- $\mathrm{Paillier}\ [1999]$ ---is semantically secure under the Decisional Composite Residuosity assumption; i.e., assuming that given an RSA modulus n and an integer z, it is hard to decide whether z is an n-residue modulo n^2 or not.

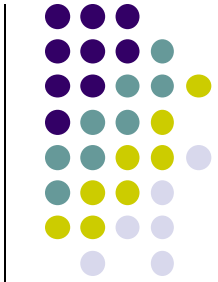
  At this, $\mathrm{Enc}_{g,n}(m) = g^m\, r^n \bmod n^2$, so $\mathrm{Enc}(a+b) = \mathrm{Enc}(a)\cdot\mathrm{Enc}(b)$

$$\mathrm{Enc}(r_i\, P(s_i) + s_i) = \mathrm{Enc}(P(s_i))^{r_i} \cdot \mathrm{Enc}(s_i) = [\Pi_1^k\, \mathrm{Enc}(\alpha_j)^{s_i^j}]^{r_i}\, \mathrm{Enc}(s_i)$$

Indeed it works!

- We get:
  - Correctness: once $\mathrm{Interaction}$ is executed, with overwhelming probability, C gets as private output the intersection $C \cap S$ and $|S|$. √
  - Server Privacy: the execution of $\mathrm{Interaction}$ leaks no information about $S \cap C^c$ (apart from its size) √ - as Paillier is semantically secure, the encryptions $Enc(r_i, P(s_i) + s_i)$ with s_i not in the intersection look like randomly selected integers mod n^2
  - Client Privacy: no information about C, apart from its size v, is leaked from $\mathrm{Interaction}$. √ -- again, comes from the fact that Paillier is semantically secure.
- Note however both v and w are leaked by $\mathrm{Interaction}$

Extensions



- Size Hiding: two-side, if an upper bound M on the sizes of both client and server´s set is known (only of interest if M<<< | U |)
- Malicious Insiders

"Bounded"- Freedman et al

Setup: run by Client, selects an encoding of U as a subset of Z_n^* ,
runs a key generation algorithm for Paillier encryption which outputs (s_k, p_k)

p_k, Encoding

Interaction:

$C = \{c_1, \ldots, c_v\}, s_k$

$S = \{s_1, \ldots, s_w\}$,
$\pi \in_R S_w$

$P(x) = x^{(M-v)} \Pi_1^{v} (x - c_i)$
$= \Sigma_0^{M} \alpha_i t^i$

$\{ Enc(\alpha_i) \}_{i=0}^{M}$

"Bounded"- Freedman et al

Setup: run by Client, selects an encoding of U as a subset of Z_n^* ,
runs a key generation algorithm for Paillier encryption which outputs (s_k, p_k)

p_k, Encoding

Interaction:

$C = \{c_1, \ldots, c_v\}$, s_k

$S = \{s_1, \ldots, s_w\}$,
$\pi \in_R S_w$

$P(x) = x^{(M-v)} \Pi_1^{v} (x - c_i)$
$= \Sigma_0^{M} \alpha_i t^i$

$\{ Enc(\alpha_i) \}_{i=0}^{M}$ →

for i=1 to w
$r_i \in_R Z_n^*$
$e_i = Enc(r_i P(s_i) + s_i)$

← $\{ \pi(e_i) \}_{i=1}^{M}$

for i=w+1 to M
$e_i \in_R Z_n^2$

"Bounded"- Freedman et al

Setup: run by Client, selects an encoding of U as a subset of Z_n^* ,
runs a key generation algorithm for Paillier encryption which outputs (s_k, p_k)

p_k, Encoding

Interaction:

$C = \{c_1,\ldots, c_v\}, s_k$

$S = \{s_1,\ldots, s_w\}$

$\pi \in_R S_w$

$P(x) = x^{(M-v)}\Pi_1^{v} (x- c_i)$
$= \Sigma_0^{M} \alpha_i t^i$

$\{ Enc(\alpha_i) \}_{i=0}^{M}$ →

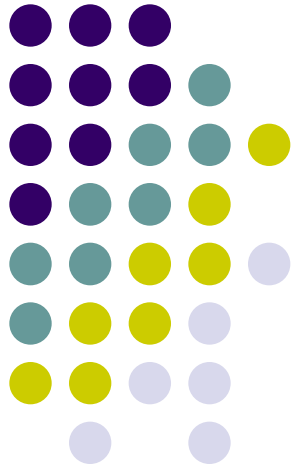
for i=1 to w
$r_i \in_R Z_n^*$
$e_i = Enc(r_i P(s_i) + s_i)$

← $\{ \pi(e_i) \}_{i=1}^{M}$

for i=w+1 to M
$e_i \in_R Z_n^2$

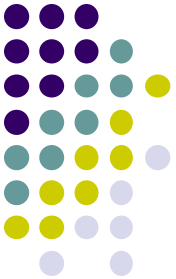
$C \cap S := \{c_1,\ldots, c_v\} \cap \{ Dec(e_i) \}_{i=1}^{M}$

Size-Hiding PSI: towars practical constructions



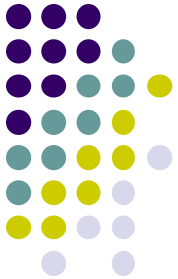
M.I. Glez. Vasco
Marzo 2015

One-side Size Hiding



- RSA construction in ROM, Ateniese et al.[2011]
- Construction from fully-homomorphic encryption, , Orlandi et al [2013]
- What if you want it more efficient? : STRONG RSA based construction using hash proof systems [In progress..]

Tools/assumptions



- Strong RSA assumption:

  given N, y, it is computationally hard to produce e, z with $z^e = y \bmod N$.

- Smooth projective hashing, $\mathrm{Cramer\ et\ al},\ [2002]$:

  Let X, Π, S be non-empty sets, $L \subseteq X$, and K a finite index set.
  Consider $F:=\{ F_k : X \mapsto \Pi \}_{k \in K}$ and $\alpha : K \mapsto S$.

Then the tuple **F** = (F, K, X, L, Π, S, α) is a *smooth projective hash family* whenever there are efficient algorithms for computing F_k (x) whenever

- either (x, k) is given as an input
- or (x, w, α(k)) is given as an input, where w is a "witness" for $x \in L$

Smoothness → Given only x and α(k), F_k(x) is "indistinguishable" from random

(our ussage involves also a special type of commitment schemes….)

D'Arco et al: Setup

S executes a key generation process for RSA encryption. As a result, she knows two primes p, q and constructs $N = pq$, with $p = 2p'+1$, $q = 2q'+1$.

She also selects a suitable pseudorandom function H and a hash proof system F, as well as g a generator of QR_N

N, g, H, F

$C = \{c_1, \ldots, c_v\}$

$S = \{s_1, \ldots, s_w\}$

$hc_i = H(c_i), \ i=1,\ldots,v$

For $i=1$ to v,

$PCH_i = \Pi_{l \neq i, l=1}^{v} hc_l$

$PCH = \Pi_{l=1}^{v} hc_l, \ A = g^{PCH} \bmod N$

Select random key k for F

$hs_j = H(s_j), \ j=1,\ldots,w$

D'Arco et al : Interaction

$C = \{c_1, \ldots, c_v\}$

$S = \{s_1, \ldots, s_w\}$

$R_c \in_R \{1, \ldots, N^2\}$, $X = A^{Rc} \bmod N$ X

D'Arco et al : Interaction

$C = \{c_1, \ldots, c_v\}$

$S = \{s_1, \ldots, s_w\}$

$R_c \in_R \{1, \ldots, N^2\}$, $X = A^{Rc} \bmod N$ X

For j=1 to w,

$y_j = X^{(1/hs_j)} \bmod N =$

$= g^{PCH\ Rc\ (1/hs_j)}$

Select random nonces $r_j, \ldots r_w$

$comm_j = comm((h_j, y_j), r_j)$

$f_j = f_k((hs_j, y_j), comm(y_j, r_j))$

Z, $\{t_1, \ldots, t_w\}$

compute projection keys

$\alpha(k, comm_j)$

set $t_j = (comm_j, r_j, \alpha(k, comm_j))$

D'Arco et al : Interaction

$C = \{c_1, \ldots, c_v\}$

$S = \{s_1, \ldots, s_w\}$

$R_c \in_R \{1, \ldots, N^2\}$, $X = A^{Rc} \bmod N$ — X →

For j=1 to w,

$y_j = X^{(1/hs_j)} \bmod N =$

$= g^{PCH\ Rc\ (1/hs_j)}$

Select random nonces $r_j, \ldots r_w$

$comm_j = comm((h_j, y_j), r_j)$

$\phi_j = f_k(hs_j, y_j), comm(y_j, r_j))$

compute projection keys

$\alpha(k, comm_j)$

← Z, $\{t_1, \ldots, t_w\}$

set $t_j = (comm_j, r_j, \alpha(k, comm_j))$

For i=1 to v,

$z_i = Z^{RcPCHi} \bmod N =$

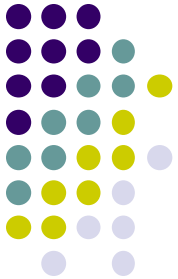
$= g^{PCHiRc} \bmod N$

For j=1 to w,

if $f_k((hc_i, z)_i, comm(y_j, r_j)) = \phi_j$

$c_i \in C \cap S$

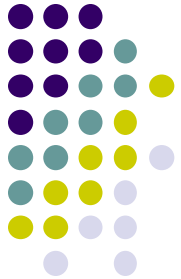
D'Arco et al.



- We get:
  - Correctness: once $\mathrm{Interaction}$ is executed, with overwhelming probability, C gets as private output the intersection $C \cap S$ and |S|

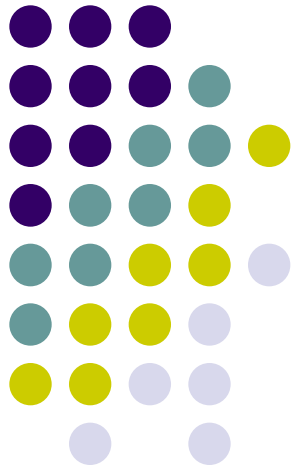
    √ - this comes from the fact that H is pseudorandom: collision-free/division intractable
  - Server Privacy: the execution of $\mathrm{Interaction}$ leaks no information about $S \cap C^c$ (yet w is leaked!) √ - (otherwise, the server would violate strong RSA assumption via Shamir´s trick..)
  - Client Privacy: no information about C, is leaked from $\mathrm{Interaction}$ √ -- x is statistically indistinguishable from a random element in QR_N.

Open Problems



- A OPE-like size hiding construction from algebraic curves (a bittersweet story so far…)
- Considering multiple executions (unlinkability)
- Two –sided: efficient uncondicionally secure (with TTP), and computationally secure (we do have one which uses the whole ground set)
- Boosting efficiency (in particular, for the case of Malicious Insiders)
- Multisets

¡Gracias!



M.I. Glez. Vasco
Marzo 2015