How to Tell Which of the Encrypted Numbers is Greater

Vladimir Kolesnikov

Bell Laboratories
Murray Hill, New Jersey, USA

Joint work with Ian F. Blake (University of Toronto)

Contents

Motivation

HAHA!! I'll set
y := x – 0.01

A: I would like to buy those sleek Matrix sunglasses.

B: My prices are so low, I cannot tell them!
Tell me how much money you have (x), and if it's more than my price (y), I'd sell it to you for y.

A: We better securely evaluate Greater Than (GT).

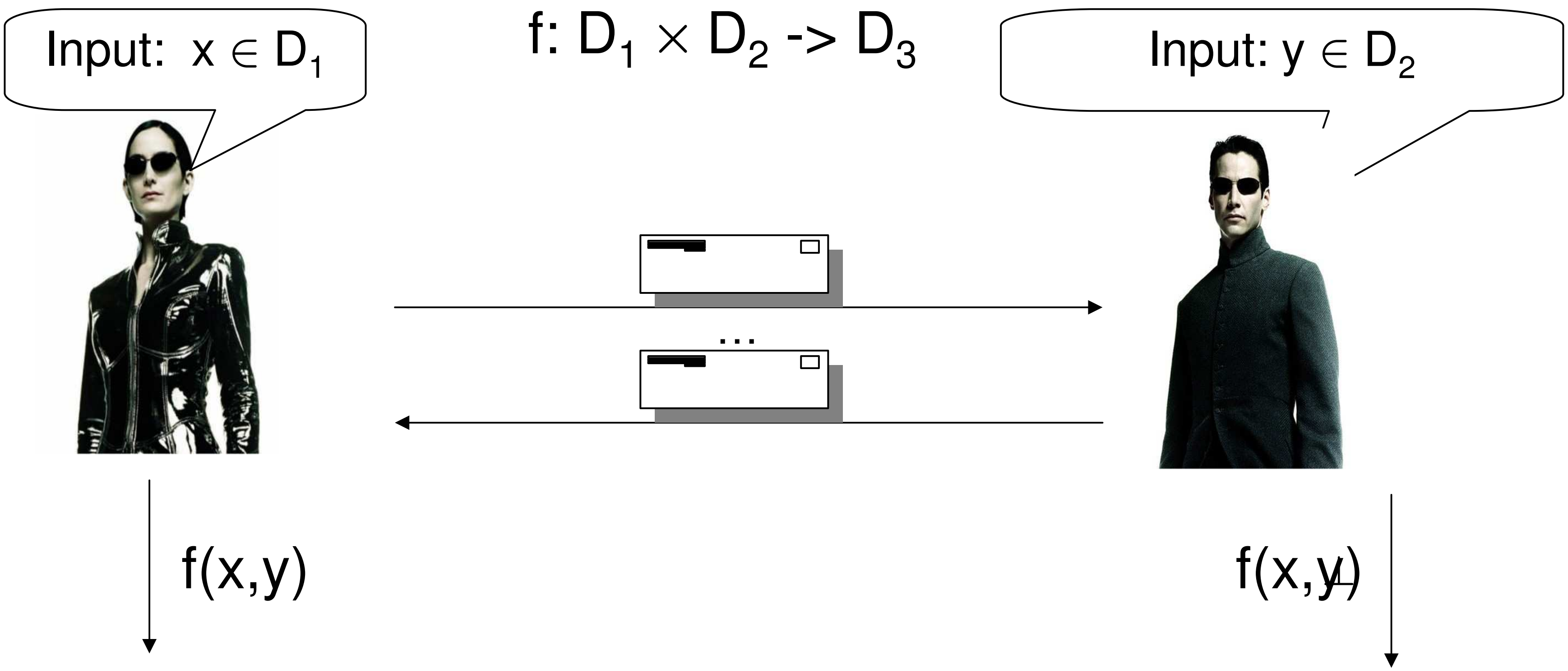
GT Uses:

Auction systems, bargaining
Secure database mining
Sales of electronic goods

Secure Function Evaluation (SFE)

- When you don't trust your partner
- Parties want to evaluate a function F on their inputs, but keep inputs private.
- Assume secure channels between parties
- Large research effort

One-Round
Secure Function Evaluation



SFE Models

- Semi-honest
  - Both players follow the protocol
  - Observe communication, try to learn additional info
- Malicious
  - Players can freely cheat
  - Solutions can be obtained by “compilation” of a semi-honest protocol

One Round SFE

- Reduces opportunities for Alice to cheat
  - Can only substitute input and misinterpret output
  - Great when asymmetric trust among parties
    - E.g. Bank and Client
- Reduces latencies
- Some applications require non-interactivity
  - Auctions
  - Mobile agents
  - Computing on Encrypted Data

One Round SFE

x

y

Enc(x) →

← Enc(encode(GT(x,y)))

↓ GT(x,y)

GT(x,y) →

Homomorphic encryption

Enc(x), Enc(y) can compute Enc (x+y)

What if Alice lies about the output?

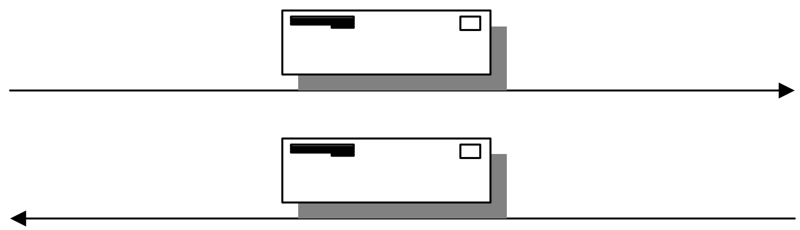
Idea: output values (0,1) are sent with authenticators

Strong Conditional OT (SCOT)

Input: x

Predicate Q(x,y)

Input: y, secrets s_0, s_1



Learn: $s_{Q(x,y)}$

Learn: nothing

Tool: Additively Homomorphic Encryption

Encryption scheme, such that:

Given $E(m_1)$, $E(m_2)$ and public key,
allows to compute $E(m_1 + m_2)$

We use scheme with large plaintext group.

The Paillier scheme satisfies our requirements

Can compute $E(cm_1 + m_2)$ from c, $E(m_1)$, $E(m_2)$

The GT-SCOT Protocol

$x_1, \ldots, x_n$

pub, pri

$x_1, \ldots, x_n$ pub

$s_0, s_1, y_1, \ldots, y_n$

$x_1, \ldots, x_n$ pub

$d = x_1 - y_1, \ldots, x_n - y_n$

$x \oplus y = (x-y)^2 = x - 2xy + y$

$f = x_1 \oplus y_1, \ldots, x_n \oplus y_n$

$f = 0\ 0\ 1\ 0\ 0\ 1\ \ 1\ \ \ 0 \ldots$

$\gamma: \gamma_0 = 0,\ \gamma_i = 2\gamma_{i-1} + f_i$

$\gamma = 0\ \ 0\ 0\ 1\ 2\ 4\ 9\ \ 19\ \ 38 \ldots$

$\delta:\ \delta_i = d_i + r_i\,(\gamma_i - 1)$

$\gamma - 1 = -1\ {-1}\ 0\ 1\ 3\ 8\ \ 18\ \ 37 \ldots$

$\mu: \mu_i = \frac{1}{2}\left((s_1 - s_0)\delta_i + s_1 + s_0\right)$

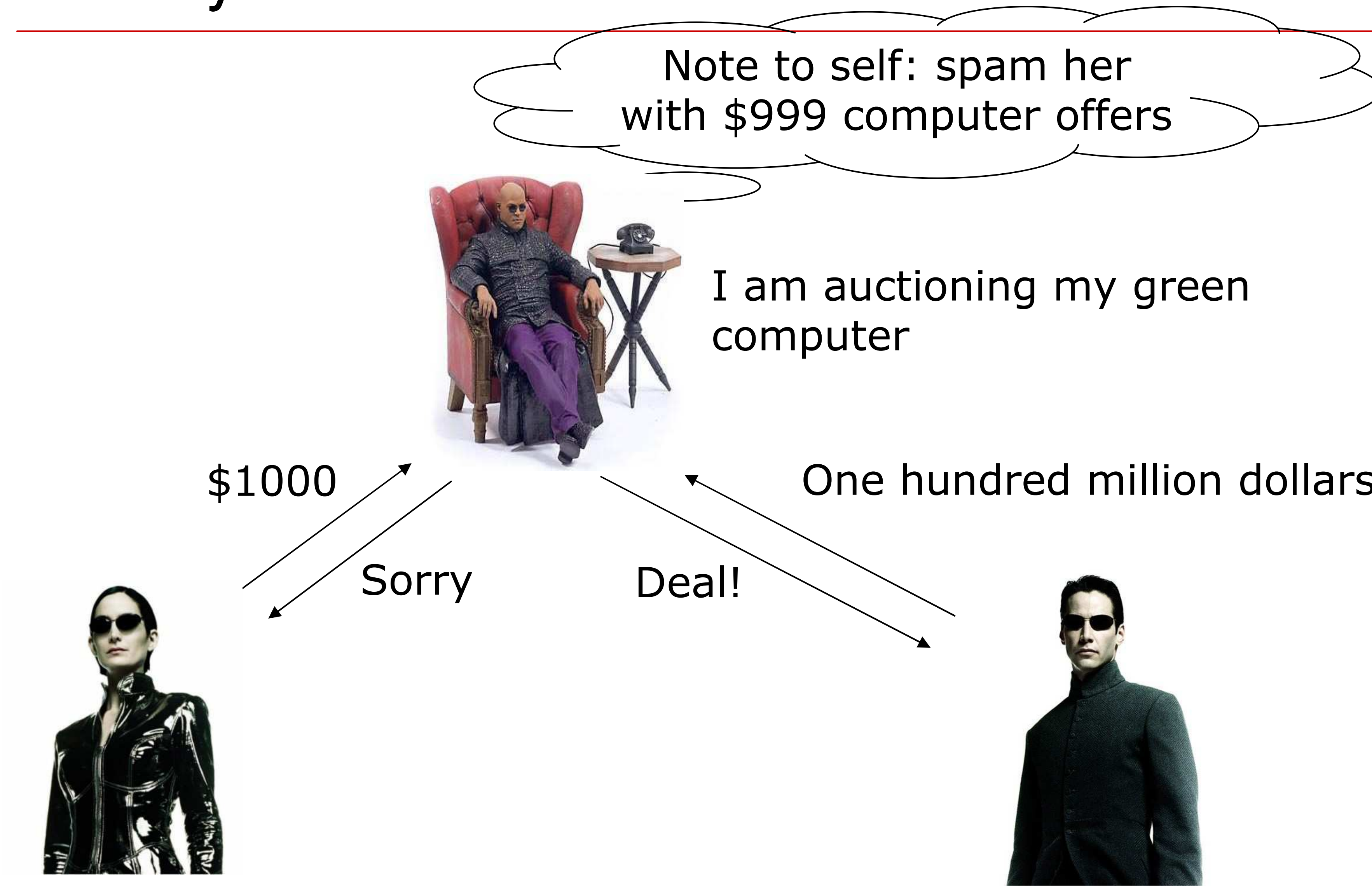
$r\,(\gamma - 1) = r_1 r_2\, 0\ r_3\ r_4 r_5\ \ r_6\ \ \ r_7 \ldots$

$d + r\,(\gamma - 1) = t_1\ t_2\ d_i\ t_3\ t_4 t_5\ \ t_6\ \ \ t_7 \ldots$

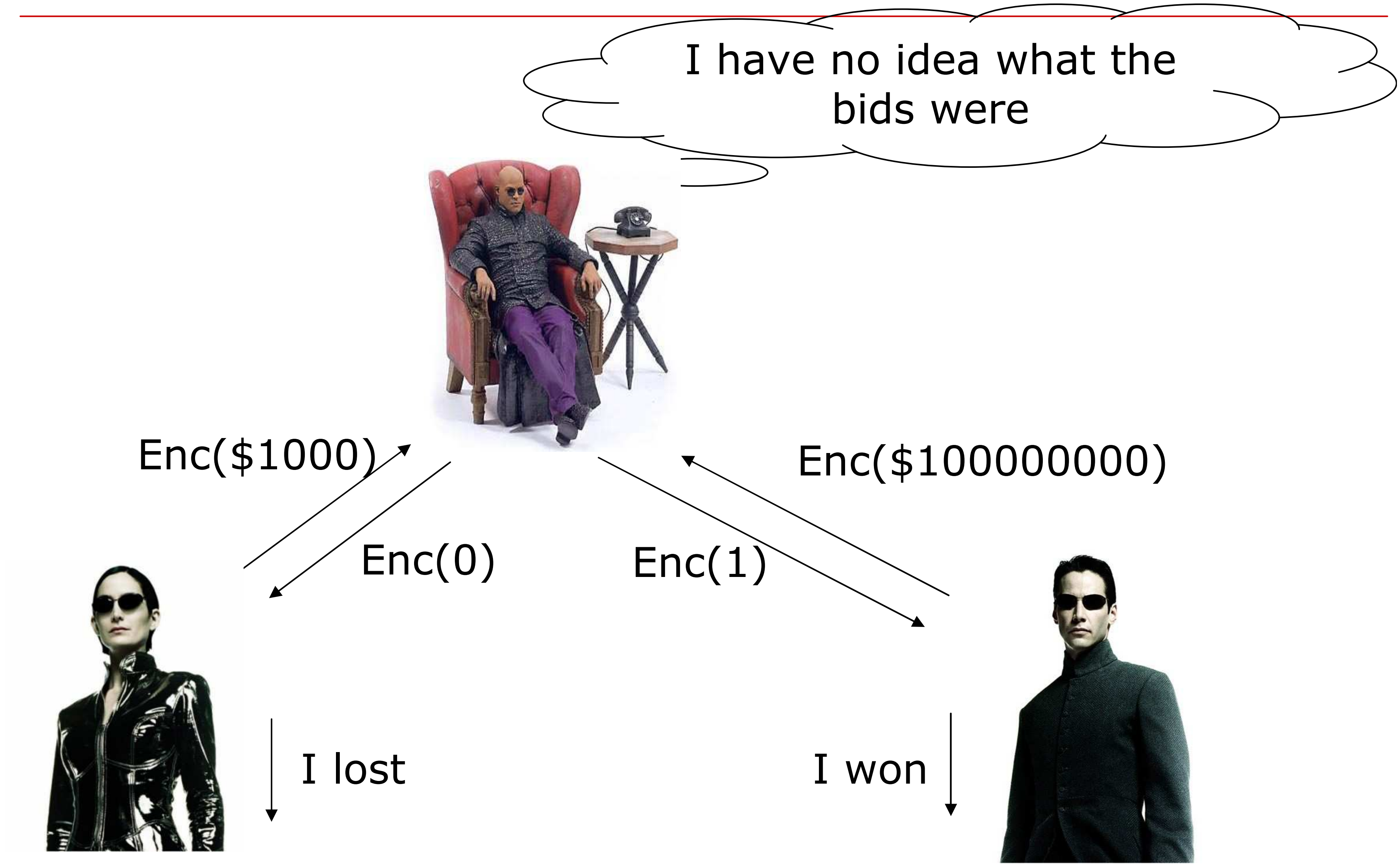
$\pi(\mu)$ $\longleftarrow$ $\pi(\mu)$

$\downarrow s_j$

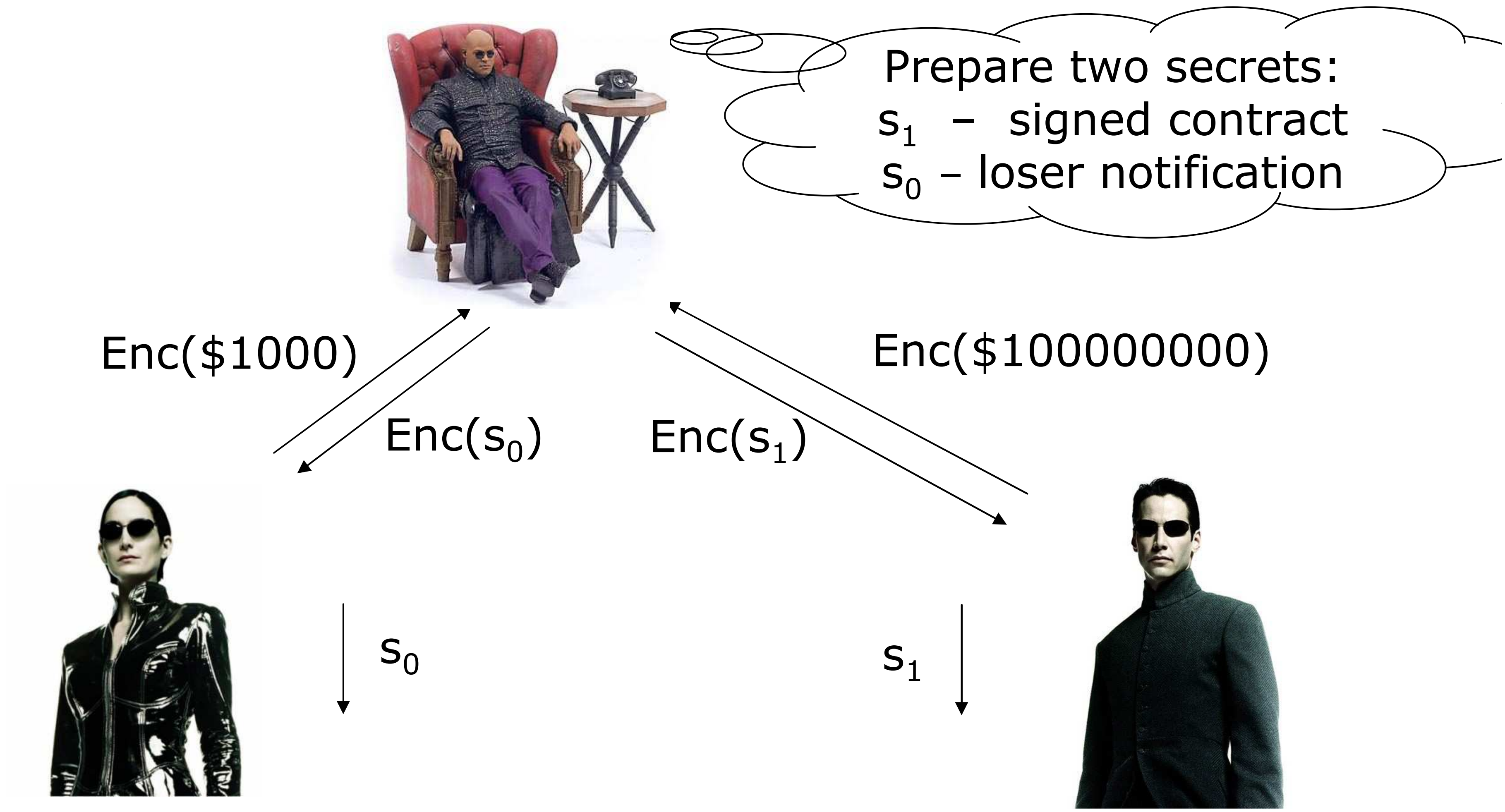
Privacy in Auctions



Comparing Encrypted Numbers



Conditional Encrypted Mapping (CEM)



Q-CEM

$Q(x,y)$ s_0, s_1

$m = Rmap(s_0, s_1, e_0, e_1, pk)$

$e_0 = Enc(x)$

$e_1 = Enc(y)$

?

m

$Rec(m, sk) = s_{Q(x,y)}$

?

Pair (Rmap, Rec) for Q is a Q-CEM

Definitional Choices

CEM: Rmap(s_0, s_1, e_0, e_1, pk), Rec(m, sk)

Strong notion of privacy

- Output of Rmap contains no statistical information other than the value $s_{Q(x,y)}$
  - Strong composability
- Holds for all generated key pairs, valid inputs and randomness used in encryption
  - E.g. Adv does not benefit from maliciously choosing randomness when encrypting inputs

Definitional Choices

CEM: Rmap(s_0, s_1, e_0, e_1, pk), Rec(m, sk)

Do not specify security requirements of the encryption scheme

- One definition is useable in most settings
- Delay discussion of easy but tedious details (e.g. what if inputs contain decryption keys)
- Q-CEM with semantically secure encryption gives a protocol in the semi-honest model
    - can be modified to withstand malicious players (ZK or the light-weight CDS)

The GT-CEM Construction

x

$x_1, \ldots, x_n$

s_0, s_1

y

$y_1, \ldots, y_n$

$d = 0\ 0\ {}^{1}_{-1}\ 0\ 0\ {}^{1}_{-1}\ {}^{1}_{-1}\ 0 \ldots$

$d = x_1 - y_1, \ldots, x_n - y_n$

$f = 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0 \ldots$

$f = x_1 \oplus y_1, \ldots, x_n \oplus y_n$

$\gamma = 0\ c\ c\ {-1}\ c_1\ c_2\ c_3\ c_4\ c_5 \ldots$

γ: $\gamma_0 = 0$, $\gamma_i = r_i \gamma_{i-1} + d_i$

$x \oplus y = (x-y)^2 = x - 2xy + y$

Linear Map

$0 \to R$ | $0 \to R$
$-1 \to s_0$ | $-1 \to ES_0$
$1 \to s_1$ | $1 \to ES_1$

ES_i is a randomized encoding of s_i

- contains no other information

Randomized Mapping

Given s_0, s_1

ES_0, ES_1, $f(x) = ax + b$

$f(-1) = b\text{-}a = ES_0$ (1)

$f(1) = a+b = ES_1$ (2)

$f(0) = b = ½\,(ES_0 + ES_1) = R'$

Assume s_0, s_1 contain redundancy

Choose $R \in_R Z_N$. View R as blocks r_0, r_1: $R = r_0\, 2^k + r_1$

ES_0 = s_0 / r_0 . r_1 — r_0 . s_0

ES_1 = r_0 . s_1 / r_1 — s_1 . r_1

c=0 — $c \in_R \{0,1\}$ — c=1

Set f = ax+b to satisfy (1),(2)

- $f(-1)$, $f(1)$ contain s_0, s_1 and no extra information*
- $f(0) = ½\,(ES_0 + ES_1) = ½\,(s_0\, 2^k + r_1 + r_0\, 2^k + s_1) =$
  $½\,(R + \ldots\,) = R'$

Application: Purchasing Movies (Aiello, Ishai, Reingold 2001)

Open an account. Pay \$y

Enc(y)

Buy movie x for \$x.

Enc(x)

If $x < y$, send K_X

Resource Comparison

Factor nc or λc improvement in communication.
Similar improvement in computation.

Protocol	Comparable Modular Multiplications			Communication	Comment
	client	server	total		
F01	$4nc\lambda\nu$	$24nc\lambda$	$32nc\lambda + 4nc\lambda\nu$	$4nc\lambda\nu$	
D00	$8n^2c\nu$	$12n^2c$	$12n^2c + 8n^2c\nu$	$8n^2c\nu$	
Our work	$16n\nu$	$16n\nu$	$32n\nu$	$2n\nu$	$c < \nu/2 - \lambda$

c-bit secrets are transferred based on comparison of n-bit numbers.
λ and ν are the correctness and security parameter

Summary

- Define several basic primitives
  - Strong Conditional Oblivious Transfer
  - Conditional Encrypted Mapping
- Give new efficient *Greater-Than* protocols
- Papers available online

Questions?