Strong Conditional Oblivious Transfer and Computing on Intervals

Vladimir Kolesnikov
Joint work with Ian F. Blake

University of Toronto

Motivation for the Greater Than Predicate

HAHA!! I’ll set
$y := x – 0.01$

A: I would like to buy tickets to Cheju Island.

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
- Secure database mining
- Computational Geometry

Previous work on GT

- Yao's Two Millionaires
- Yao's Garbled Circuit
  - Rogaway, 1991
  - Naor, Pinkas, Sumner, 1999
  - Lindell, Pinkas, 2004
- Sander, Young, Yung, 1999
- Fischlin, 2001
- Many others

Our Model

A: Let’s do it in **one round** – I hate waiting!

B: Let’s be **Semi-Honest**.
That means we will not deviate from our protocol. We can, however, try to learn things we aren’t supposed to by observing our communication.

A: Also, I will have **unlimited computation power**.

B: That sounds complicated. Most efficient solutions won’t work (e.g. garbled circuit).

Tools – Homomorphic Encryption

Encryption scheme, such that:

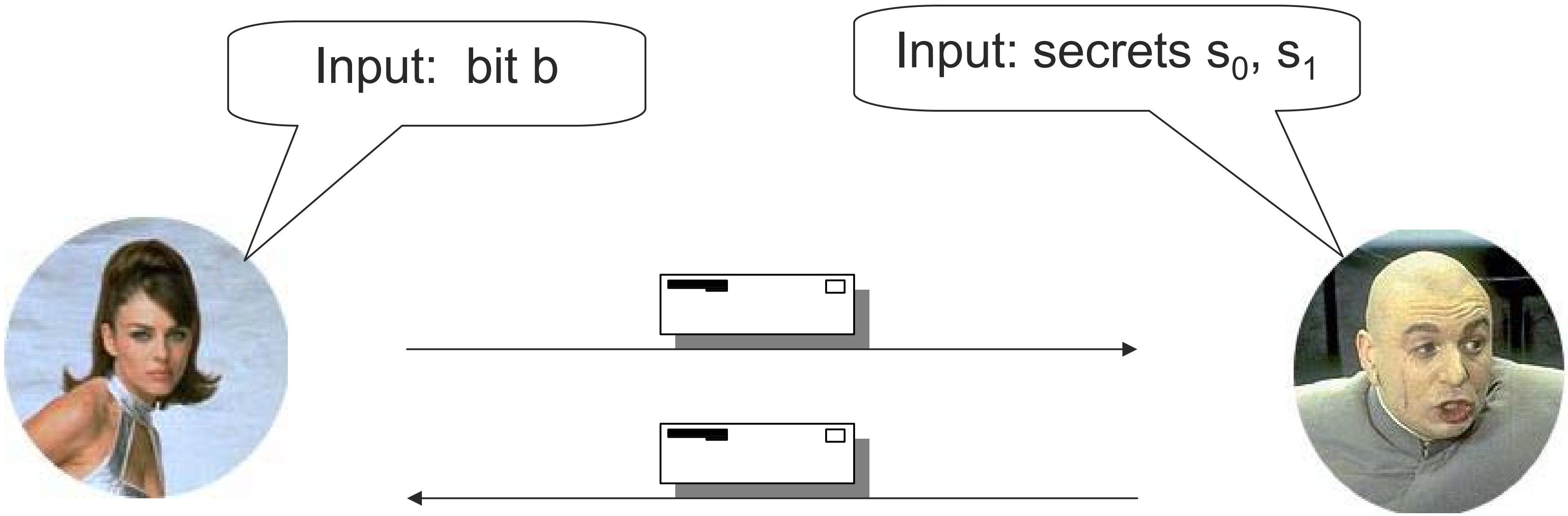
Given $E(m_1)$, $E(m_2)$ and public key,
allows to compute $E(m_1 \otimes m_2)$

We will need:

- Additively homomorphic ($\otimes = +$) schemes
- Large plaintext group

The Paillier scheme satisfies our requirements

Oblivious Transfer (OT)



Learn: s_b

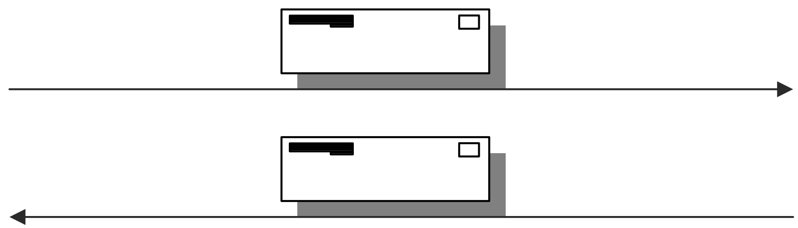
Learn: nothing

Strong Conditional OT (SCOT)

Input: x

Predicate Q(x,y)

Input: y, secrets s_0, s_1



Learn: $\mathbf{s}_{Q(x,y)}$

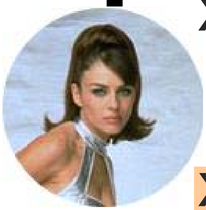
Learn: nothing

Q-SCOT

Is a generalization of:

- COT of Di Crescenzo, Ostrovsky, Rajagopalan, 1999
- OT
- Secure evaluation of $Q(x,y)$

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

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

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

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

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

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

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

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

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

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

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

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

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

$\downarrow s_j$

Interval-SCOT

x

$x_1, x_2, s_0, s_1 \in D_S$

$s_0 = a_1+b_1 = a_2+b_2$

$s_1 = a_2+b_1$

GT-SCOT($a_1|a_2$? $x<x_1$)

GT-SCOT($b_1|b_2$? $x<x_2$)

$\downarrow a_i+b_j$

Union of Intervals-SCOT

$I_1,\ldots, I_k, s_0, s_1 \in D_S$

x

$s_1 = \sum_i s_{i1}$

$s_1 - s_0 = s_{i1} - s_{i0}$

I-SCOT($s_{11}|s_{10}$? $x \in I_1$)

I-SCOT($s_{k1}|s_{k0}$? $x \in I_k$)

$\downarrow \sum_i s_{i?}$

Conclusions

- General and composable definition of SCOT
- SCOT solutions (GT, I, UI)
  - Simple and composable
  - Orders of magnitude improvement in communication (loss in computational efficiency in some cases)
  - Especially efficient for transferring larger secrets (e.g. $\approx$1000 bits)

Resource Comparison

Protocol	GT predicate		c-bit GT-SCOT, c<log N		k-UI-SCOT	
	mod. mult.	comm.	mod. mult.	comm.	mod. mult.	comm.
F01	$8n\lambda$	$\lambda n \log N$	$32nc\lambda$	$4nc\lambda \log N$	$64kn\lambda^2$	$8kn\lambda^2 \log N$
DOR99	$8n$	$4n \log N$	N/A	N/A	N/A	N/A
our work	$16n \log N$	$4n \log N$	$20n \log N$	$4n \log N$	$40kn \log N$	$8kn \log N$