RSAConference2020

San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: **CRYP-R02**

Improved Secure Integer Comparison via Homomorphic Encryption

Jacques Traoré

Research Engineer
Orange Labs
Caen, France
Joint work with Florian Bourse and Olivier Sanders

#RSAC

Agenda

- Secure Integer Comparison
- Related Works
- Our Contribution
- Efficiency
- Conclusion

RSAConference2020

Secure Integer Comparison

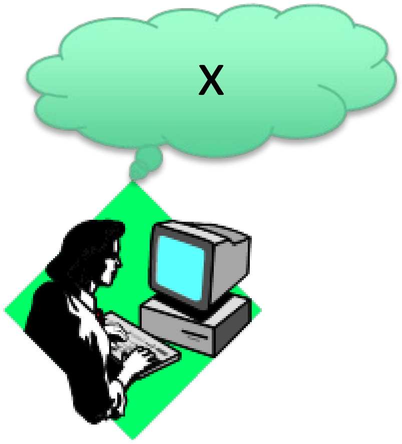
Yao's Millionaires' Problem

Alice

Bob

How can they find out **who is richer** without revealing their **actual wealth**?

Secure Integer Comparison



Alice

Bob

How to determine whether $X \geq Y$ or not without revealing anything more about X and Y?

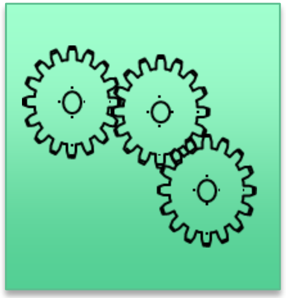
FHE Setting

- **Inputs of A:** a pair of keys (SK, PK) of a public key encryption scheme E
- **Inputs of B:** two ciphertexts $C_1 = E_{PK}$ (x) and $C_2 = E_{PK}$ (y)

Party B

C_1

C_2



Homomorphic computations

$C = E_{PK}$ (b)

- **Goal of B:** to blindly compute an encryption C of the boolean $(x < y)$ under PK
- **Applications:** e.g. «Machine Learning Classification over Encrypted Data » (NDSS 2015)

RSAConference2020

Related Works

Related Works

- **FOCS 1982**: based on « **Garbled Circuits** » $\Rightarrow$ rather important communication complexity
- **CT-RSA 2001:** based on **Homomorphic Encryption**. Involves bitwise encryption of the integers $\Rightarrow$ a complexity of at least $log_2(M)$ operations where M is the bound on the integers to compare
- **CT-RSA 2018**: based on an **Homomorphic Threshold Encryption scheme**. Allows to directly compare small integers but at the cost of more interactions between the two parties
- **Crypto 2018**: based on **FHE**. Only supports a bounded message space which has to be defined at setup time $\Rightarrow$ works well but on very small sized inputs.
- **CT-RSA 2019**: based on the « **Legendre Symbol** ». Unfortunately can only handle integers of limited size. Seems difficult to extend it to support large inputs

Our Contribution

Our Contribution

- We propose **two protocols** that respectively improve CEK (**CT-RSA 2018**) and BMMP (**Crypto 2018**)
- For both of our protocols, we **avoid binary decomposition** in order to improve the performances
- Compared to CEK we managed to **divide** by **two** the **number of interactions** between the two protagonists
- Compared to BMMP, our new protocol allows to securely compare **large** (a priori **unbounded**) **integer**

Encryption Scheme

- Based on CEK

$$E_{PK}(m) = g^{b^m} h^r \bmod N$$

- Public Key PK= $\{N, b, d, g, h, u\}$
  - $N = pq$ where $p = 2b^d p_s p_t + 1$ and $q = 2b^d q_s q_t + 1$
  - g is of order b^d, h is of order $p_s q_s$ and u is a an upper bound on $p_s q_s$
- Private key $SK = p_s q_s \hat{s}$ where $\hat{s} = (p_s q_s)^{-1} \bmod b^d$
- Message space $\mathcal{M}$= $\{0, \dots, d-1\}$

Decryption

- To decrypt a ciphertext $\boldsymbol{C}$ using a private key $\boldsymbol{SK}$

$$
\begin{aligned}
(C)^{sk} \bmod N &= \left(g^{b^m} h^r\right)^{p_s q_s \hat{s}} \\
&= \left(g^{b^m}\right)^{p_s q_s \hat{s}} \left(h^r\right)^{p_s q_s \hat{s}} \\
&= g^{b^m p_s q_s \hat{s}} \\
&= g^{b^m}
\end{aligned}
$$

- Find m by exhaustive search. Easy provided d is a small integer

Threshold Homomorphic Property

$$C = E_{PK}(x)^{b^{d-y}} = (g^{b^x} h^r)^{b^{d-y}}$$
$$= g^{b^x b^{d-y}} h^{r'}$$
$$= g^{b^{d+(x-y)}} h^{r'}$$

If $\boldsymbol{x - y \geq 0}$ then $b^{d+(x-y)} \equiv 0 \bmod b^d$

$$C = E_{PK}(x)^{b^{d-y}} = g^0 h^{r'}$$

$$D_{SK}(C) = \emptyset$$

Threshold Homomorphic Property (II)

$$C = E_{PK}(x)^{b^{d-y}} = g^{b^{d+(x-y)}} h^{r'}$$

If $x - y \geq 0$ then $D_{SK}(C) = \emptyset$ and we obtain no other information on y

If $x - y < 0$ then $D_{SK}(C) = d + (x - y)$

- An interesting property for secure integer comparison provided we find a way to **blind the integer y** in particular **when $x - y < 0$**

Protocol for the Millionaires' Problem

Alice (sk, x)		Bob (Pk, y)
		Choose random: u, v, r_2
$C \leftarrow E_{PK}(x) = g^{b^x} h^{r_1}$	$\xrightarrow{\quad C \quad}$	
		Compute: $D \leftarrow C^{ub^{d-y}} g^v h^{r_2} = g^{ub^{d+(x-y)}+v} h^{r_3}$
		$D' = \mathcal{H}(g^v)$
Compute:	$\xleftarrow{\quad D, D' \quad}$	
$C' = D^{sk}$		
If $D' = \mathcal{H}(C')$		
Return $(x \geq y)$		
Return $(x < y)$ **otherwise**		

Security

- We proved the security for both A and B against **honest-but-curious adversaries** in the **random oracle model**
- **Privacy for A**. We show that B learns nothing about x during the protocol

  Theorem. Under the **Small RSA Subgroup Decision Assumption**, B's view is computationally indistinguishable from a uniformly random element in $\mathbb{QR}_N$ for any message x

- **Privacy for B**. We show that A only learns the output of the protocol $(x \geq y)$ and nothing else about y

  Theorem. There exists an efficient simulator S, such that $S(1^\lambda, (x \geq y))$ is statistically indistinguishable from A's view for any messages x and y in the random oracle model

FHE based Secure Integer Comparison

- A variant of Bourse et al. FHE scheme (Crypto 2018)
- Our scheme supports a **non-binary message space**:

  Let B be an integer. The message space will be $\mathcal{M} = \{-B+1, \ldots, B-1\}$

- Ciphertexts can be **homomorphically added** and **scaled** by a known integer constant.

> Roughly: let $c_1 = E_{PK}(m_1)$ and $c_2 = E_{PK}(m_2)$ and $w \in \mathbb{Z}$
>
> $$D_{SK}(c_1 + wc_2) = m_1 + wm_2$$

- **Ternary Sign computation:**

> **Input**: $c = E_{PK}(m)$ where $m \in \{-B+1, \ldots, B-1\}$
>
> **Output**: $c' = E_{PK}(s)$ where $s = \begin{cases} -1, & m < 0 \\ 0, & m = 0 \\ 1, & m > 0 \end{cases}$

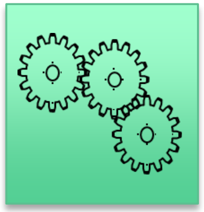
A Protocol for Small Integers

Let x and y be two integers in $\mathfrak{B} = \{0, \ldots, B-1\}$

- **Inputs of B:** $C_1 = E_{PK}(x)$ and $C_2 = E_{PK}(y)$
- Compute $C = E_{PK}(x - y)$ using the homomorphic properties satisfied by E

Party B

$C_1 \rightarrow$, $C_2 \rightarrow$

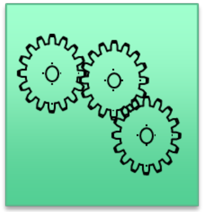


Homomorphic computations

$\rightarrow C = E_{PK}(x - y)$

- **Output:** $C' = E_{PK}(s)$ an encryption of s, the ternary sign of $(x - y)$

$C \rightarrow$



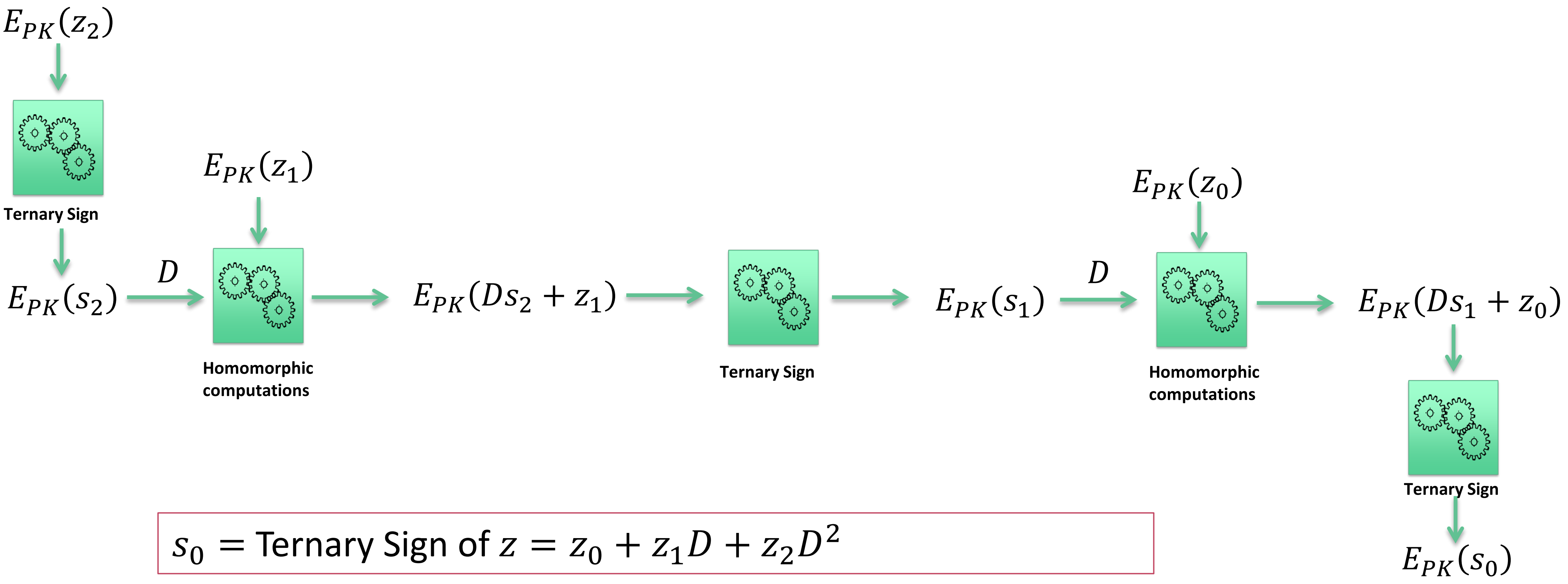
Ternary Sign computations

$\rightarrow C'$

A Protocol for Large Integers

Let $x = x_0 + x_1 D + x_2 D^2$ and $y = y_0 + y_1 D + y_2 D^2$ where $D = \frac{B}{2}$

- **Inputs of B:** $C_1 = (E_{PK}(x_0), E_{PK}(x_1), E_{PK}(x_2))$ and $C_2 = (E_{PK}(y_0), E_{PK}(y_1), E_{PK}(y_2))$
- Compute $C = (E_{PK}(z_0), E_{PK}(z_1), E_{PK}(z_2))$ where $z_i = x_i - y_i$ using the homomorphic properties of E



$\boxed{s_0 = \text{Ternary Sign of } z = z_0 + z_1 D + z_2 D^2}$

RSAConference2020

Efficiency

Efficiency

- There is a wide range of solutions to the Millionaires' problem from garbled circuits to homomorphic encryption

- Compared to similar solutions based on homomorphic encryption such DGK (ACISP'07) and CEK (CT-RSA 2018) our protocol is:
  - 4 times faster than DGK for a 256-bit security level
  - in two-passes instead of 4 with CEK and do not require a Plaintext Equality TEST (PET)

- Our **FHE** solution allows to compare 32 bits integers (on a Core i7-3630QM laptop) in 1023ms on a single core, 297ms on 8 cores and 165ms with maximum parallelization
  - Greater than comparison of 32 bits integers with Kolesnikov et al. protocol (CANS 09) would require 4224 ms on the same laptop

Conclusion

Conclusion

- We have introduced two new solutions to the **Millionaires' problem** in **two different settings**
- Our first solution leverages the **homomorphic encryption scheme of Carlton et al**. to construct a **two-passes integer comparison protocol** that improves over the state of the art.
- Our second solution extends the **FHE construction of Bourse et al**. to enable efficient computation of the encrypted boolean $(x \leq y)$ given only the encryption of (a priori unbounded) integers x and y
- Both solutions share the same guiding principles, namely **reducing** as much as possible the **number of interactions** and **avoiding bitwise decomposition** of the integers