Communication complexity of the forge-and-lose technique
(@ secure evaluation of AES-128 and SHA-256 circuits)

Luís Brandão+

University of Lisbon / LaSIGE (Portugal) and Carnegie Mellon University (USA)

Crypto 2013 Rump Session
(Santa Barbara, USA, August 20)

+ Ph.D. student, supported by the Portuguese Foundation for Science and Technology (FCT) through the Carnegie Mellon Portugal Program under Grant SFRH/BD/33770/2009.
S2PC via cut-and-choose of garbled circuits
S2PC via cut-and-choose of garbled circuits
S2PC via cut-and-choose of garbled circuits

S2PC (ideal):

Alice → TTP

Usual C&C methods

Here are 123 garbled circuits

OK, let me verify that 74 are OK, and then I’ll evaluate the other 49
S2PC via cut-and-choose of garbled circuits

S2PC (ideal):

Alice $x$ TTP $y$ Bob

Usual C&C methods

Here are 123 garbled circuits

GC1 GC2 ... GC123

OK, let me verify that 74 are OK, and then I’ll evaluate the other 49

The output is OK only if majority of evaluation GCs is correct
S2PC via cut-and-choose of garbled circuits

S2PC (ideal):

Alice $x$ \rightarrow TTP $y$ \rightarrow Bob $C(x,y)$

Usual C&C methods

Here are 123 garbled circuits

$GC_1 \quad GC_2 \quad \cdots \quad GC_{123}$

OK, let me verify that 74 are OK, and then I’ll evaluate the other 49

$Pr_{error} \approx 1.26 \cdot 2^{-0.32 \text{ s}}$

The output is OK only if majority of evaluation GCs is correct
S2PC via cut-and-choose of garbled circuits

S2PC (ideal):

Alice $x$ TTP $y$ Bob

Usual C&C methods

Here are 123 garbled circuits

GC$_1$ GC$_2$ \ldots GC$_{123}$

OK, let me verify that 74 are OK, and then I’ll evaluate the other 49

$Pr_{\text{error}} \approx 1.26 \cdot 2^{-0.32s}$

The output is OK only if majority of evaluation GCs is correct

$s \equiv \# \text{GCs}$
S2PC via cut-and-choose of garbled circuits

S2PC (ideal):

Usual C&C methods

Here are 123 garbled circuits

OK, let me verify that 74 are OK, and then I’ll evaluate the other 49

Alice

Bob

TTP

Pr_{error} \approx 1.26 \cdot 2^{-0.32s}

Example: 123 GCs for Pr_{error} < 2^{-40}

The output is OK only if majority of evaluation GCs is correct
**S2PC via cut-and-choose of garbled circuits**

**S2PC (ideal):**

Alice $\rightarrow$ $x$ TTP $\rightarrow$ Bob $\leftarrow$ $y$ $\rightarrow$ $C(x,y)$

**Usual C&C methods**

Here are 123 garbled circuits

$\text{GC}_1 \quad \text{GC}_2 \quad \cdots \quad \text{GC}_{123}$

OK, let me verify that 74 are OK, and then I’ll evaluate the other 49

Pr$_{\text{error}} \approx 1.26 \cdot 2^{-0.32s}$

The output is OK only if majority of evaluation GCs is correct

**New optimal(?) C&C methods in 2013**

$\text{GC}_1 \quad \text{GC}_2 \quad \cdots \quad \text{GC}_{123}$

The output is OK if at least one evaluation GC is correct

Example: 123 GCs for Pr$_{\text{error}} < 2^{-40}$

$s \equiv \# \text{ GCs}$
**S2PC via cut-and-choose of garbled circuits**

**S2PC (ideal):**

Alice \( \rightarrow \) TTP \( \rightarrow \) Bob

\( x \rightarrow \) TTP \( \rightarrow \) Bob \( \rightarrow \) C(x,y)

**Usual C&C methods**

Here are 123 garbled circuits

\( \text{GC}_1 \quad \text{GC}_2 \quad \cdots \quad \text{GC}_{123} \)

OK, let me verify that 74 are OK, and then I’ll evaluate the other 49

\( s \equiv \# \text{GCs} \)

\( \Pr_{\text{error}} \approx 1.26 \cdot 2^{-0.32s} \)

Example: 123 GCs for \( \Pr_{\text{error}} < 2^{-40} \)

**New optimal(?) C&C methods in 2013**

<table>
<thead>
<tr>
<th>C&amp;C proportions (verify vs. evaluate)</th>
<th>( \Pr_{\text{error}} )</th>
<th># GCs: ( \Pr_{\text{error}} \leq 2^{-40} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed</td>
<td>( \approx 1.25 \cdot 2^{-s + (\log_2 s)/2} )</td>
<td>44</td>
</tr>
<tr>
<td>Variable</td>
<td>( 2^{-s} )</td>
<td>40</td>
</tr>
</tbody>
</table>

The output is OK only if

- majority of evaluation GCs is correct

The output is OK if

- at least one evaluation GC is correct

© 2012-2013 Luís Brandão
S2PC via cut-and-choose of garbled circuits

**S2PC (ideal):**

- **Alice** sends $x$ to **TTP**, who sends $y$ to **Bob**.
- **TTP** computes $C(x,y)$.

**Usual C&C methods**

Here are 123 garbled circuits: $\text{GC}_1, \text{GC}_2, \ldots, \text{GC}_{123}$.

**Example:** 123 GCs for $\text{Pr}_{\text{error}} < 2^{-40}$

$$\text{Pr}_{\text{error}} \approx 1.26 \cdot 2^{-0.32s}$$

The output is OK only if the majority of evaluation GCs is correct.

**New optimal (?) C&C methods in 2013**

<table>
<thead>
<tr>
<th>C&amp;C proportions (verify vs. evaluate)</th>
<th>$\text{Pr}_{\text{error}}$</th>
<th>$# \text{GCs}$: $\text{Pr}_{\text{error}} &lt; 2^{-40}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed</td>
<td>$\approx 1.25 \cdot 2^{-s} + \frac{(\log_2 s)}{2}$</td>
<td>44</td>
</tr>
<tr>
<td>Variable</td>
<td>$2^{-s}$</td>
<td>40</td>
</tr>
</tbody>
</table>

The output is OK if at least one evaluation GC is correct.

Compare against 123
Three new optimal(?) C&C methods
Three new optimal(?) C&C methods

- [Lin13] – crypto 2013 (Wednesday)
Three new optimal(?) C&C methods

- [Lin13] – crypto 2013 (Wednesday)
- [HKE13] – crypto 2013 (Wednesday)
Three new optimal(?) C&C methods

- [Lin13] – crypto 2013 (Wednesday)
- [HKE13] – crypto 2013 (Wednesday)
- [Bra13] – crypto 2013 rump session
Three new optimal(?) C&C methods

- [Lin13] – crypto 2013 (Wednesday)
- [HKE13] – crypto 2013 (Wednesday)
- [Bra13] – crypto 2013 rump session (now)
Three new optimal(?) C&C methods

- [Lin13] – crypto 2013 (Wednesday)
- [HKE13] – crypto 2013 (Wednesday)
- [Bra13] – crypto 2013 rump session (now)

Forge-and-lose technique

\[ GC_1 \text{ output wire } i \rightarrow \text{ wire key for bit 1} \]

\[ GC_2 \text{ output wire } i \rightarrow \text{ wire key for bit 0} \]
Three new optimal(?) C&C methods

- [Lin13] – crypto 2013 (Wednesday)
- [HKE13] – crypto 2013 (Wednesday)
- [Bra13] – crypto 2013 rump session (now)

Forge-and-lose technique

Bob

Correct?

FORGED?

wire key for bit 1

wire key for bit 0

GC\textsubscript{1} output wire \textit{i}

GC\textsubscript{2} output wire \textit{i}
Three new optimal(?) C&C methods

- [Lin13] – crypto 2013 (Wednesday)
- [HKE13] – crypto 2013 (Wednesday)
- [Bra13] – crypto 2013 rump session (now)

Forge-and-lose technique

Bob

Correct?

FORGED?

Wire key for bit 1

Connect

Encoding of 1

Wire key for bit 0

Connect

Encoding of 0

(group elements)
Three new optimal(?) C&C methods

- [Lin13] – crypto 2013 (Wednesday)
- [HKE13] – crypto 2013 (Wednesday)
- [Bra13] – crypto 2013 rump session (now)

Forge-and-lose technique
Three new optimal(?) C&C methods

- [Lin13] – crypto 2013 (Wednesday)
- [HKE13] – crypto 2013 (Wednesday)
- [Bra13] – crypto 2013 rump session (now)

Forge-and-lose technique

Correct?

Bob

 Trapdoor (Decryption key)

\( \text{GC}_1 \) output wire \( i \)

wire key for bit 1

Encoding of 1

(decommitments of same BitCom)

\( \text{GC}_2 \) output wire \( i \)

wire key for bit 0

Encoding of 0

(group elements)
Three new optimal(?) C&C methods

- [Lin13] – crypto 2013 (Wednesday)
- [HKE13] – crypto 2013 (Wednesday)
- [Bra13] – crypto 2013 rump session (now)

Forge-and-lose technique

- Correct?
- FORGED?
- (decommitments of same BitCom)
- (group elements)
- (early in protocol)

Bob

Trapdoor

(Decryption key)

Encryption

of

input of Alice

Encryption

of

for bit 0

Connect

for bit 1

Connect

GC1 output wire i

GC2 output wire i

Encoding

of 1

Encoding

of 0

wire key

wire key

Correct?

FORGED?
Three new optimal(?) C&C methods

- [Lin13] – crypto 2013 (Wednesday)
- [HKE13] – crypto 2013 (Wednesday)
- [Bra13] – crypto 2013 rump session (now)

Forge-and-lose technique

Three new optimal(?) C&C methods

- [Lin13] – crypto 2013 (Wednesday)
- [HKE13] – crypto 2013 (Wednesday)
- [Bra13] – crypto 2013 rump session (now)

Forge-and-lose technique
Three new optimal(?) C&C methods

- [Lin13] – crypto 2013 (Wednesday)
- [HKE13] – crypto 2013 (Wednesday)
- [Bra13] – crypto 2013 rump session (now)

Forge-and-lose technique

1. Alice's input
2. Trapdoor (Decryption key)
3. Evaluate Boolean circuit
4. Output of Bob

- Correct?
- FORGED?

<table>
<thead>
<tr>
<th>GC1 output wire i</th>
<th>GC2 output wire i</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wire key for bit 0</td>
<td>Wire key for bit 1</td>
</tr>
</tbody>
</table>

Encoding of 1
- (decommitments of same BitCom)

Encoding of 0
- (group elements)

Encrypted input of Alice
- (early in protocol)

Bob's input
- Alice's input

LOSES privacy

FORGED?
- GC1 output wire i
- GC2 output wire i

Connect

Correct?
- GC1 output wire i
- GC2 output wire i

Bob's input
- Alice's input

Evaluate Boolean circuit
- Output of Bob

© 2012-2013 Luís Brandão

“Communication Complexity of the forge-and-lose technique”

Crypto 2013 Rump Session – Santa Barbara, US, August 20, 2013
Three new optimal(?) C&C methods

- [Lin13] – crypto 2013 (Wednesday)
- [HKE13] – crypto 2013 (Wednesday)
- [Bra13] – crypto 2013 rump session (now)

Forge-and-lose technique

\[ \text{Correct?} \quad \text{Bob} \]

\[ \text{Output wire } i \quad \text{GC}_1 \]

\[ \text{Wire key for bit 1} \quad \text{Encoding of 1} \quad \text{(decommitments of same BitCom)} \]

\[ \text{FORGED?} \quad \text{GC}_2 \]

\[ \text{Wire key for bit 0} \quad \text{Encoding of 0} \quad \text{(group elements)} \]

Trapdoor (Decryption key)

Alice’s input

Bob’s input

Evaluate Boolean circuit

Encrypted input of Alice (early in protocol)

Output of Bob

Alice

LOSES privacy
Three new optimal(?) C&C methods

- [Lin13] – crypto 2013 (Wednesday)
- [HKE13] – crypto 2013 (Wednesday)

Forge-and-lose technique

- Alice sends a Trapdoor (Decryption key) to Bob.
- Bob connects his input to the Trapdoor and evaluates a Boolean circuit.
- If the output of the circuit is correct, Bob decrypts the wire key and compares it with his input.
- If the output is FORGED, Bob loses privacy.
- Alice’s input is connected to the circuit and Bob’s input is connected to the circuit as well.
- Encryption of 0 and 1 are connected to the circuit.
- (decommitments of same BitCom) and (group elements) are connected to the circuit.
- The output of the Bob is encrypted input of Alice.
- The output of the circuit is connected to the output of Bob.

© 2012-2013 Luís Brandão

“Communication Complexity of the forge-and-lose technique”

Crypto 2013 Rump Session – Santa Barbara, US, August 20, 2013
Benchmarking communication in F&L
Benchmarking communication in F&L

- Crypto security: 128 bits → 3,072-bit Blum integers [NIST-SP800-57]
- Statistical security: 40 bits ($\Pr_{\text{error}} \leq 2^{-40}$)
- Garbled gates: 384 bits
- Symmetric commitments: 256 / 384 bits
Benchmarking communication in F&L

- Crypto security: 128 bits → 3,072-bit Blum integers [NIST-SP800-57]
- Statistical security: 40 bits ($\Pr_{\text{error}} \leq 2^{-40}$)
- Garbled gates: 384 bits
- Symmetric commitments: 256 / 384 bits

<table>
<thead>
<tr>
<th></th>
<th>AES-128</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>C_A</td>
</tr>
<tr>
<td>$l_A=l_B=l_B'$</td>
<td>128</td>
</tr>
<tr>
<td>$(l_A+l_B+l_B')/</td>
<td>C_A</td>
</tr>
<tr>
<td>s (# GCs)</td>
<td>41 123</td>
</tr>
<tr>
<td>Max # evaluation GCs</td>
<td>20 8</td>
</tr>
<tr>
<td>RSC@GCs</td>
<td>no yes</td>
</tr>
<tr>
<td>GCs (Mb)</td>
<td>107 21</td>
</tr>
<tr>
<td>Total (Mb)</td>
<td>161 55</td>
</tr>
<tr>
<td>Overhead from non-GCs (%)</td>
<td>50% 163%</td>
</tr>
</tbody>
</table>
## Benchmarking communication in F&L

- Crypto security: 128 bits $\rightarrow$ 3,072-bit Blum integers [NIST-SP800-57]
- Statistical security: 40 bits ($\Pr_{\text{error}} \leq 2^{-40}$)
- Garbled gates: 384 bits
- Symmetric commitments: 256 / 384 bits

<table>
<thead>
<tr>
<th></th>
<th>AES-128</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>C_A</td>
</tr>
<tr>
<td>$l_A=l_B=l'_B$</td>
<td>128</td>
</tr>
<tr>
<td>$(l_A+l_B+l'_B)/</td>
<td>C_A</td>
</tr>
<tr>
<td>s (# GCs)</td>
<td>41</td>
</tr>
<tr>
<td>Max # evaluation GCs</td>
<td>20</td>
</tr>
<tr>
<td>RSC@GCs</td>
<td>no</td>
</tr>
<tr>
<td>GCs (Mb)</td>
<td>107</td>
</tr>
<tr>
<td>Total (Mb)</td>
<td>161</td>
</tr>
<tr>
<td>Overhead from non-GCs (%)</td>
<td>50% 163%</td>
</tr>
</tbody>
</table>

- Benchmarking AES-128
Benchmarking communication in F&L

- Crypto security: 128 bits → 3,072-bit Blum integers [NIST-SP800-57]
- Statistical security: 40 bits (Pr_{error} \leq 2^{-40})
- Garbled gates: 384 bits
- Symmetric commitments: 256 / 384 bits

<table>
<thead>
<tr>
<th>AES-128</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>C\rangle$ [Bri13]</td>
</tr>
<tr>
<td>$l_A=l_B=l'_B$</td>
<td>128</td>
</tr>
<tr>
<td>$(l_A+l_B+l'_B)/</td>
<td>C\rangle$</td>
</tr>
<tr>
<td>s (# GCs)</td>
<td>41</td>
</tr>
<tr>
<td>Max # evaluation GCs</td>
<td>20</td>
</tr>
<tr>
<td>RSC@GCs</td>
<td>no</td>
</tr>
<tr>
<td>GCs (Mb)</td>
<td>107</td>
</tr>
<tr>
<td>Total (Mb)</td>
<td>161</td>
</tr>
<tr>
<td>Overhead from non-GCs (%)</td>
<td>50%</td>
</tr>
</tbody>
</table>
Benchmarking communication in F&L

- Crypto security: 128 bits → 3,072-bit Blum integers [NIST-SP800-57]
- Statistical security: 40 bits ($\text{Pr}_{\text{error}} \leq 2^{-40}$)
- Garbled gates: 384 bits
- Symmetric commitments: 256 / 384 bits

<table>
<thead>
<tr>
<th></th>
<th>AES-128</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>C_{\lambda}</td>
</tr>
<tr>
<td>$l_{A}=l_{B}=l'_{B}$</td>
<td>128</td>
</tr>
<tr>
<td>$(l_{A}+l_{B}+l'_{B})/</td>
<td>C_{\lambda}</td>
</tr>
<tr>
<td>s (# GCs)</td>
<td>41</td>
</tr>
<tr>
<td>Max # evaluation GCs</td>
<td>20</td>
</tr>
<tr>
<td>RSC@GCs</td>
<td>no</td>
</tr>
<tr>
<td>GCs (Mb)</td>
<td>107</td>
</tr>
<tr>
<td>Total (Mb)</td>
<td>161</td>
</tr>
<tr>
<td>Overhead from non-GCs (%)</td>
<td>50%</td>
</tr>
</tbody>
</table>
Benchmarking communication in F&L

- Crypto security: 128 bits → 3,072-bit Blum integers [NIST-SP800-57]
- Statistical security: 40 bits (Pr_{error} ≤ 2^{-40})
- Garbled gates: 384 bits
- Symmetric commitments: 256 / 384 bits

<table>
<thead>
<tr>
<th></th>
<th>AES-128</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>C_A</td>
</tr>
<tr>
<td>(l_A=l_B=l'_B</td>
<td>128</td>
</tr>
<tr>
<td>((l_A+l_B+l'_B)/</td>
<td>C_A</td>
</tr>
<tr>
<td>s (# GCs)</td>
<td>41</td>
</tr>
<tr>
<td>Max # evaluation GCs</td>
<td>20</td>
</tr>
<tr>
<td>RSC@GCs</td>
<td>no</td>
</tr>
<tr>
<td>GCs (Mb)</td>
<td>107</td>
</tr>
<tr>
<td>Total (Mb)</td>
<td>161</td>
</tr>
<tr>
<td>Overhead from non-GCs (%)</td>
<td>50%</td>
</tr>
</tbody>
</table>
Benchmarking communication in F&L

- Crypto security: 128 bits → 3,072-bit Blum integers [NIST-SP800-57]
- Statistical security: 40 bits ($\text{Pr}_{\text{error}} \leq 2^{-40}$)
- Garbled gates: 384 bits
- Symmetric commitments: 256 / 384 bits

<table>
<thead>
<tr>
<th></th>
<th>AES-128</th>
<th>SHA-256</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>C_{\lambda}</td>
<td>$ [Bri13]</td>
</tr>
<tr>
<td>$l_A=l_B=l'_B$</td>
<td>128</td>
<td>256</td>
</tr>
<tr>
<td>$(l_A+l_B+l'_B)/</td>
<td>C_{\lambda}</td>
<td>$</td>
</tr>
<tr>
<td>s (# GCs)</td>
<td>41</td>
<td>123</td>
</tr>
<tr>
<td>Max # evaluation GCs</td>
<td>20</td>
<td>8</td>
</tr>
<tr>
<td>RSC@GCs</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>GCs (Mb)</td>
<td>107</td>
<td>21</td>
</tr>
<tr>
<td>Total (Mb)</td>
<td>161</td>
<td>55</td>
</tr>
<tr>
<td>Overhead from non-GCs (%)</td>
<td>50%</td>
<td>163%</td>
</tr>
</tbody>
</table>

© 2012-2013 Luís Brandão

“Communication Complexity of the forge-and-lose technique”

Crypto 2013 Rump Session – Santa Barbara, US, August 20, 2013
Benchmarking communication in F&L

- Crypto security: 128 bits → 3,072-bit Blum integers [NIST-SP800-57]
- Statistical security: 40 bits (Pr_{error} \leq 2^{-40})
- Garbled gates: 384 bits
- Symmetric commitments: 256 / 384 bits

<table>
<thead>
<tr>
<th></th>
<th>AES-128</th>
<th>SHA-256</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C</td>
<td>[Bri13]</td>
</tr>
<tr>
<td>l_A=l_B=l’_B</td>
<td>128</td>
<td>256</td>
</tr>
<tr>
<td>(l_A+l_B+l’_B)/</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>s (# GCs)</td>
<td>41</td>
<td>123</td>
</tr>
<tr>
<td>Max # evaluation GCs</td>
<td>20</td>
<td>8</td>
</tr>
<tr>
<td>RSC@GCs</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>GCs (Mb)</td>
<td>107</td>
<td>21</td>
</tr>
<tr>
<td>Total (Mb)</td>
<td>161</td>
<td>55</td>
</tr>
<tr>
<td>Overhead from non-GCs (%)</td>
<td>50%</td>
<td>163%</td>
</tr>
</tbody>
</table>
**Benchmarking communication in F&L**

- Crypto security: 128 bits → 3,072-bit Blum integers [NIST-SP800-57]
- Statistical security: 40 bits (Pr\(_{error}\) ≤ 2\(^{-40}\))
- Garbled gates: 384 bits
- Symmetric commitments: 256 / 384 bits

<table>
<thead>
<tr>
<th></th>
<th>AES-128</th>
<th>SHA-256</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>C_A</td>
<td>[Bri13]</td>
</tr>
<tr>
<td>(l_A=l_B=l'_B)</td>
<td>128</td>
<td>256</td>
</tr>
<tr>
<td>((l_A+l_B+l'_B)/</td>
<td>C_A</td>
<td>)</td>
</tr>
<tr>
<td>s (# GCs)</td>
<td>41</td>
<td>123</td>
</tr>
<tr>
<td>Max # evaluation GCs</td>
<td>20</td>
<td>8</td>
</tr>
<tr>
<td>RSC@GCs</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>GCs (Mb)</td>
<td>107</td>
<td>21</td>
</tr>
<tr>
<td>Total (Mb)</td>
<td>161</td>
<td>55</td>
</tr>
<tr>
<td>Overhead from non-GCs (%)</td>
<td>50%</td>
<td>163%</td>
</tr>
</tbody>
</table>

### Some aspects

- **Intractability**: Quadratic Residuosity
- **#(exps)**: O(I)
- **Oblivious Transfers**: 2-out-of-1 OT
- **Proof security**: Ideal/real simulation (with rewinding)
- **BitComs input+output**: XOR-homomorphic ⇒ Efficient linkage of S2PCs
Benchmarking communication in F&L

- Crypto security: 128 bits → 3,072-bit Blum integers [NIST-SP800-57]
- Statistical security: 40 bits (Pr_{error} ≤ 2^{-40})
- Garbled gates: 384 bits
- Symmetric commitments: 256 / 384 bits

<table>
<thead>
<tr>
<th></th>
<th>AES-128</th>
<th>SHA-256</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C_</td>
<td>[Bri13]</td>
</tr>
<tr>
<td>l_A=l_B=l_B'</td>
<td>128</td>
<td>256</td>
</tr>
<tr>
<td>(l_A+l_B+l_B')/</td>
<td>C_</td>
<td></td>
</tr>
<tr>
<td>s (# GCs)</td>
<td>41</td>
<td>123</td>
</tr>
<tr>
<td>Max # evaluation GCs</td>
<td>20</td>
<td>8</td>
</tr>
<tr>
<td>RSC@GCs</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>GCs (Mb)</td>
<td>107</td>
<td>21</td>
</tr>
<tr>
<td>Total (Mb)</td>
<td>161</td>
<td>55</td>
</tr>
<tr>
<td>Overhead from non-GCs (%)</td>
<td>50%</td>
<td>163%</td>
</tr>
</tbody>
</table>

(Further optimizations on the way)

Some aspects

- **Intractability**: Quadratic Residuosity
- #(exps): O(l)
- **Oblivious Transfers**: 2-out-of-1 OT
- **Proof security**: Ideal/real simulation (with rewinding)

**BitComs input+output:**

- XOR-homomorphic ⇒ Efficient linkage of S2PCs
Thanks

cut-and-chose
with
forge-and-lose!

lbrandao @ {fc.ul.pt, cmu.edu}