The Cut-and-Choose Game and its Application to Cryptographic Protocols

Ruiyu Zhu, Yan Huang, Jonathan Katz, abhi shelat

Indiana University

U. Maryland

Northeastern University
What is Cut-and-Choose
What is Cut-and-Choose
Applications of Cut-and-Choose

• Secure Computation
  - LP, Eurocrypt 07
  - SS, EuroCrypt 11
  - Brandão, AsiaCrypt 13
  - AMPR, Crypto14
  - Lindell, Crypto 13
  - HKE, Crypto13

• Zero-knowledge-proof
  - Blum, ICM 86

• Fair exchange of digital currency
  - BBSU, FC 12

• Secure delegation of computation
  - CKV, Crypto 10
Applications of Cut-and-Choose

- Secure Computation
  - LP, Eurocrypt 07
  - SS, EuroCrypt 11
  - Brandão, AsiaCrypt 13
  - AMPR, Crypto14
  - Lindell, Crypto 13
  - HKE, Crypto13

- Zero-knowledge-proof
  - Blum, ICM 86

- Fair exchange of digital currency
  - BBSU, FC 12

- Secure delegation of computation
  - CKV, Crypto 10
Cut-and-Choose in Secure Computation
Three Flavors of Cut-and-choose

• **SingleCut**
  – Secure if at least one evaluation-circuit is correct.
    [Lindell, Crypto 13] [HKE, Crypto 13] [Brandão, AsiaCrypt 13] [AMPR, Crypto 14]

• **MajorityCut**
  – Secure if the majority of evaluation-circuits are correct.
    [SS’ EuriCrypt 11] [LP, EuroCrypt 07] [Woodruff, EuroCrypt 07] [LP, SCN 08] [LP, JoP12]

• **BatchedCut**
  – Amortizing cost over multiple executions.
    [LR, Crypto 14] [NO, TCC09] [FJN+, EuroCrypt 13]
Three Flavors of Cut-and-choose

- **SingleCut**
  - Secure if at least one evaluation-circuit is correct.
  - Lindell, Crypto 13
  - HKE, Crypto 13
  - Brandão, AsiaCrypt 13
  - AMPR, Crypto 14

- **MajorityCut**
  - Secure if the majority of evaluation-circuits are correct.
  - SS' EuriCrypto 11
  - LP, EuroCrypt07
  - Woodruff, EuroCrypt 07
  - LP, SCN 08
  - LP, JoP12

- **BatchedCut**
  - Amortizing cost over multiple executions.
  - LR, Crypto 14
  - NO, TCC09
  - FJN+, EuroCrypt13
Expected cost:

\[
\text{checking cost} \times \frac{s}{2} + \text{evaluation cost} \times \frac{s}{2}
\]

\(s: \text{the security parameter}\)
The Cost Gap

Checking

Evaluation

Garbled Circuit

Hash

Garbled Circuit

Bandwidth Cost Ratio

$10^4 \sim 10^9$

Time Cost Ratio

$2 \sim 30$

Seed

16 bytes 32 bytes
Our Key Intuition

Evaluate *less* and check *more*.

Use *mixed-strategies*: determine the number of evaluation-circuits *probabilistically* from a custom distribution.

Use *linear programming* to find optimal parameters.
Problem Formulation

Want to minimize $\mathbb{E}[\text{cost}(r, S_{\text{eval}})]$

Subject to:

$$\text{Pr}_{\text{failure}}(S_{\text{eval}}, S_{\text{gen}}) \leq \varepsilon, \quad \forall S_{\text{gen}}$$

<table>
<thead>
<tr>
<th>$\varepsilon$</th>
<th>Upper-bound on the security failure rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r$</td>
<td>Cost ratio</td>
</tr>
<tr>
<td>$S_{\text{gen}}$</td>
<td>Generator’s strategy</td>
</tr>
<tr>
<td>$S_{\text{eval}}$</td>
<td>Evaluator’s strategy</td>
</tr>
</tbody>
</table>
Problem Formulation

Want to minimize $\mathbb{E}[\text{cost}(r, S_{eval})]$ 

Subject to:

$\Pr_{\text{failure}}(S_{eval}, S_{gen}) \leq \varepsilon, \quad \forall S_{gen}$

- $\varepsilon$: Upper-bound on the security failure rate
- $r$: Cost ratio
- $S_{gen}$: Generator’s strategy
- $S_{eval}$: Evaluator’s strategy
$S_{gen}$ and $S_{eval}$ in SingleCut

<table>
<thead>
<tr>
<th>$n$</th>
<th>The total number of circuits</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{gen}$</td>
<td>A random variable over ${0,1}^n$</td>
</tr>
<tr>
<td>$S_{eval}$</td>
<td>A random variable over ${0,1}^n$</td>
</tr>
</tbody>
</table>
$S_{gen}$ and $S_{eval}$ in SingleCut

- My only choices are which circuits to form improperly.
- I could map between binary string and strategy.

<table>
<thead>
<tr>
<th>$S_{gen}$</th>
<th>$S_{eval}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Garbled Circuits]</td>
<td>[Garbled Circuits]</td>
</tr>
<tr>
<td>[Garbled Circuits]</td>
<td>[Garbled Circuits]</td>
</tr>
<tr>
<td>[Garbled Circuits]</td>
<td>[Garbled Circuits]</td>
</tr>
</tbody>
</table>

**Failure:** $S_{gen} = S_{eval}$

<table>
<thead>
<tr>
<th>Chk</th>
<th>Eval</th>
<th>Chk</th>
<th>Eval</th>
<th>Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

So could I
Expected Cost of SingleCut

\[ \mathbb{E}[\text{cost}(r, S_{\text{eval}})] = \sum_{i=0}^{n} (ir + (n - i) \cdot 1)x_i \]

# of circuits to evaluate
Total # of circuits

\( n \quad \text{Total number of circuits} \)
\( x_i \quad \text{Probability of evaluating } i \text{ circuits} \)
Constraints on $x_i$
(because it’s a probability distribution)

\[
x_i \geq 0
\]

\[
\sum_{i=0}^{n} x_i = 1
\]

$n$  Total number of circuits

$x_i$  Probability of evaluating $i$ circuits
$$\Pr_{\text{failure}}(S_{\text{eval}}, S_{\text{gen}}) \leq \varepsilon$$

Security Holds

$$\forall S_{\text{gen}}, \Pr(S_{\text{eval}} = S_{\text{gen}}) \leq \varepsilon$$

Probability that evaluator picks any SPECIFIC strategy $a$ is bounded by $\varepsilon$.

$$\forall a \in \{0,1\}^n, \Pr(S_{\text{eval}} = a) \leq \varepsilon$$
\[ \forall a \in \{0,1\}^n, \Pr (S_{eval} = a) \leq \varepsilon \]

\[ x_i \leq \binom{n}{i} \cdot \varepsilon \]

Each pure strategy can be picked with probability at most \(\varepsilon\).

There are \(\binom{n}{i}\) pure strategies that evaluate \(i\) circuits.
Recap

Minimize:

\[
\sum_{i=0}^{n} (ir + n - i)x_i
\]

Subject to:

\[
x_i \geq 0
\]

\[
\sum_{i=0}^{n} x_i = 1
\]

\[
x_i \leq \varepsilon \binom{n}{i}
\]
Fractional Knapsack Problem

Capacity: $1/\varepsilon$ units

A greedy algorithm solves it in linear time.
Find the Best $n$

- Exhaustively search every $n$ to find the one with minimal cost.

Range of $n$

- Limitation: $n$ is publicly fixed. Followup at: https://github.com/Opt-Cut-N-Choose

Achievable with the SingleCut strategy of [Lindell, Crypto13].

Required by the security parameter $\epsilon$

$$\log_2 \frac{1}{\epsilon} \quad \frac{r + 1}{2} \log_2 \frac{1}{\epsilon}$$
Sample SingleCut Strategy for AES

<table>
<thead>
<tr>
<th>Classical Strategy</th>
<th>( n = 40 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( i )</td>
<td>( x_i ) as %</td>
</tr>
<tr>
<td>0</td>
<td>( 9.09 \cdot 10^{-11} )</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>19</td>
<td>( 11.9 \cdot 10^0 )</td>
</tr>
<tr>
<td>20</td>
<td>( 12.5 \cdot 10^0 )</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>40</td>
<td>( 9.09 \cdot 10^{-11} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Our technique</th>
<th>( n = 2267 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( i )</td>
<td>( x_i ) as %</td>
</tr>
<tr>
<td>0</td>
<td>( 9.09 \cdot 10^{-11} )</td>
</tr>
<tr>
<td>1</td>
<td>( 2.06 \cdot 10^{-7} )</td>
</tr>
<tr>
<td>2</td>
<td>( 2.34 \cdot 10^{-4} )</td>
</tr>
<tr>
<td>3</td>
<td>( 1.77 \cdot 10^{-1} )</td>
</tr>
<tr>
<td>4</td>
<td>( 99.8 \cdot 10^0 )</td>
</tr>
<tr>
<td></td>
<td>Save 77.5% b/w</td>
</tr>
</tbody>
</table>

Bandwidth cost ratio: \( r = 4533 \) For AES
Improvements on SingleCut

\[ \text{Savings} = 1 - \frac{\text{cost}_{\text{this work}}}{\text{cost}_{\text{best prior work}}} \]
Improvements on SingleCut

![Graph showing improvements on SingleCut with savings on the y-axis and cost ratio r on the x-axis. The graph highlights AES and fp-multiply.]
Formulation for MajorityCut

Minimize:
\[ \sum_{i=0}^{n} (ir + n - i)x_i \]

Subject to:
\[ x_i \geq 0 \]
\[ \sum_{i=0}^{n} x_i = 1 \]
\[ \sum_{i=0}^{\min(n,2b)} x_i \cdot \binom{n-b}{i-b} / \binom{n}{i} \leq \varepsilon \]

See the paper for details.
Sample Majority Cut Strategy

**Classical Strategy**

<table>
<thead>
<tr>
<th>$n$</th>
<th>$i$</th>
<th>$x_i$ as %</th>
</tr>
</thead>
<tbody>
<tr>
<td>124</td>
<td>43</td>
<td>100</td>
</tr>
</tbody>
</table>

**Our technique**

<table>
<thead>
<tr>
<th>$n = 175$</th>
<th>$i$</th>
<th>$x_i$ as %</th>
<th>$i$</th>
<th>$x_i$ as %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7</td>
<td>$1 \cdot 10^{-4}$</td>
<td>17</td>
<td>1.23</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>$9 \cdot 10^{-4}$</td>
<td>19</td>
<td>5.36</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>$7 \cdot 10^{-3}$</td>
<td>21</td>
<td>20.9</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>$4.54 \cdot 10^{-2}$</td>
<td>23</td>
<td>72.2</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.25</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Save 26.6% time

Time cost ratio: $r = 10$
Improvements on MajorityCut

\[ \text{Savings} = 1 - \frac{\text{cost}_{\text{this work}}}{\text{cost}_{\text{best prior work}}} \]
Improvements on MajorityCut

Savings vs Cost ratio \( r \):

- AES
- fp-multiply

Graph shows the relationship between savings and cost ratio for AES and fp-multiply operations.
Improvements on BatchedCut

$\text{Savings} = 1 - \frac{\text{cost}_{\text{this work}}}{\text{cost}_{\text{best prior work}}}$

$N$ is the size of the circuit.
Conclusion

Cut-and-choose protocols should be appropriately configured based on the security requirement and the cost ratio benchmarked at run-time.

The game solvers are available at https://github.com/cut-n-choose.

Ruiyu Zhu: zhu52@indiana.edu