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

Ruiyu Zhu, Yan Huang, Jonathan Katz, abhi shelat



Indiana University

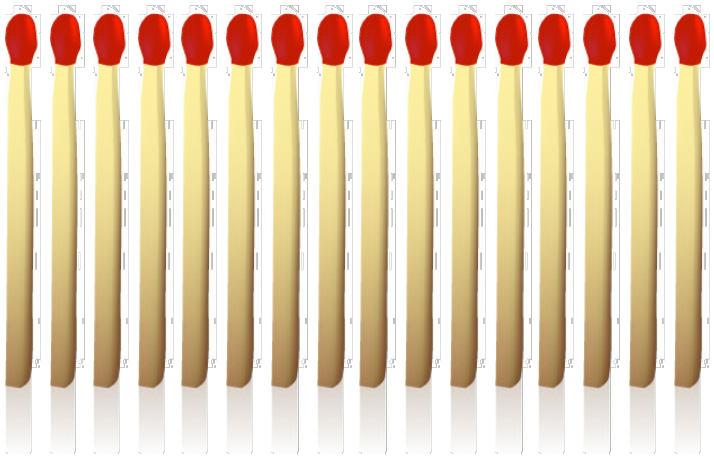


U. Maryland



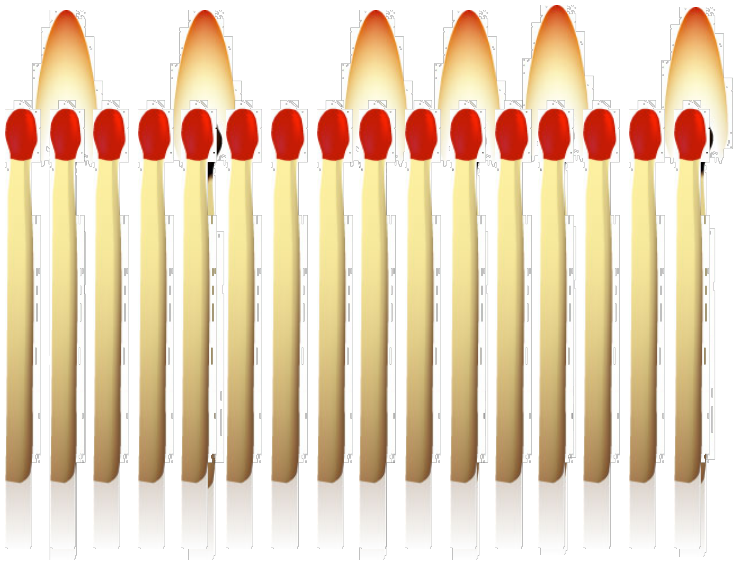
Northeastern University

What is Cut-and-Choose





What is Cut-and-Choose





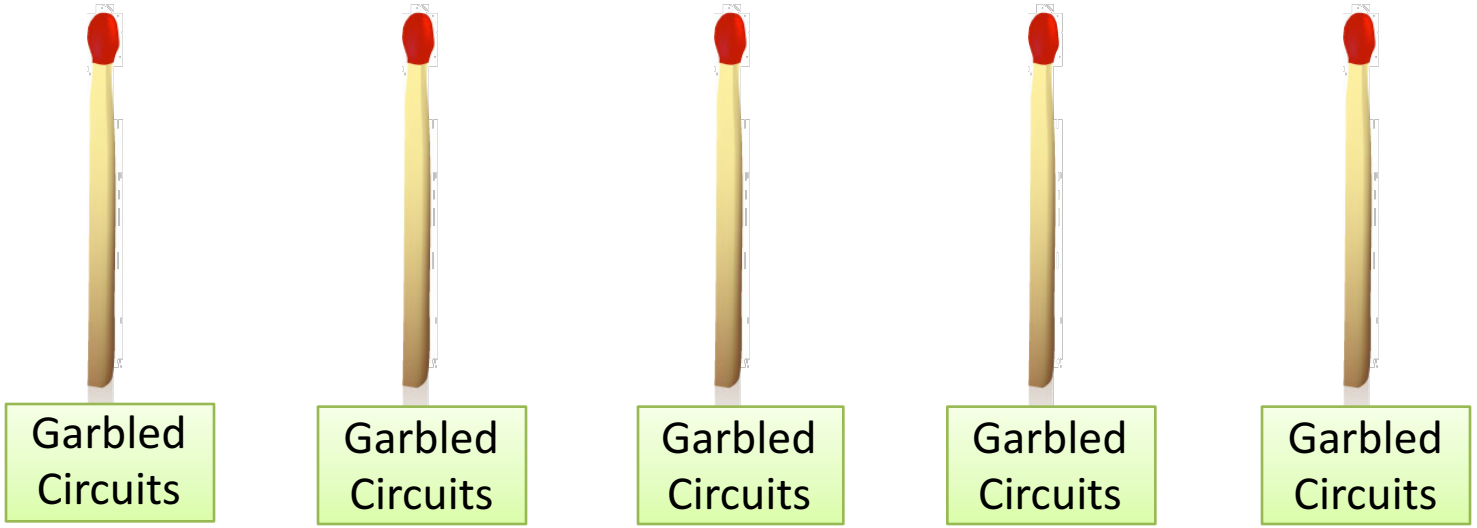
Applications of Cut-and-Choose

- 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

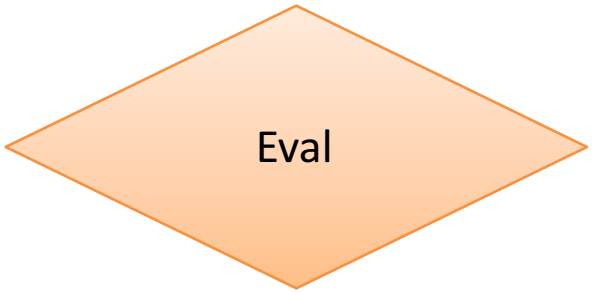
Applications of Cut-and-Choose

- 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' 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

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.

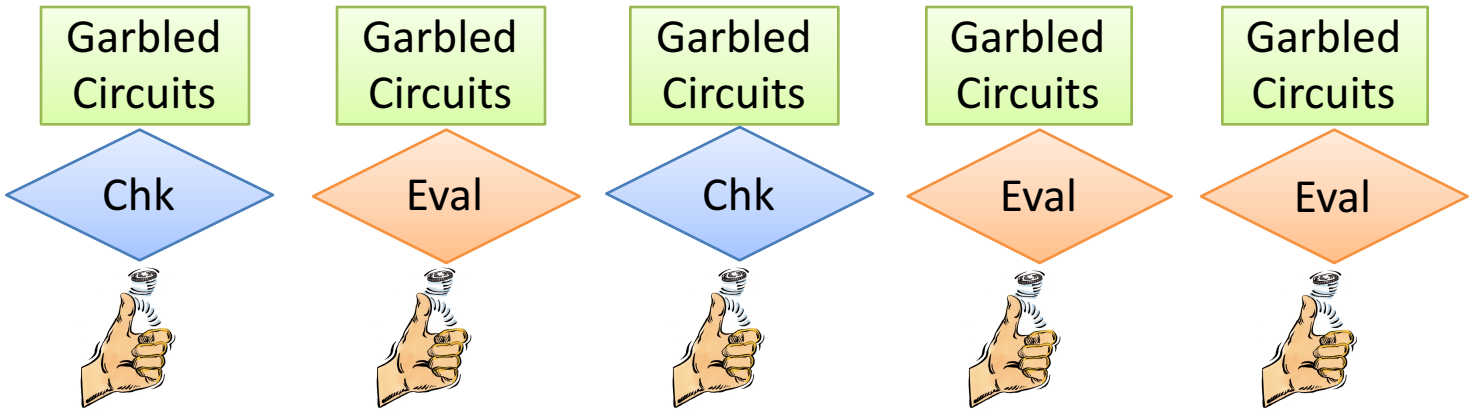
SS' EuriCrypto 11 | LP, EuroCrypt07 | Woodruff, EuroCrypt 07 | LP, SCN 08 | LP, JoP12

- BatchedCut
  - Amortizing cost over multiple executions.

LR, Crypto 14 | NO, TCC09 | FJN+, EuroCrypt13

Existing SingleCut Strategy

Lindell, Crypto 13



Expected cost:

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

s:the security parameter

The Cost Gap



Checking

Evaluation



Seed

Garbled Circuit

Hash

Garbled Circuit

Bandwidth Cost Ratio

$10^4 \sim 10^9$

Time Cost Ratio

$2 \sim 30$



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

Subject to:

"For all cheating strategies"

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

ε	Upper-bound on the security failure rate
r	Cost ratio
S_{gen}	Generator's strategy
S_{eval}	Evaluator's strategy

Problem Formulation

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

Subject to:

"For all cheating strategies"

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

ε	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

n	The total number of circuits
S_{gen}	A random variable over $\{0,1\}^n$
S_{eval}	A random variable over $\{0,1\}^n$

S_{gen} and S_{eval} in SingleCut

My only choices are which circuits to form improperly.



I could map between binary string and strategy

0 1 0 1 1

Garbled Circuits | Garbled Circuits | Garbled Circuits | Garbled Circuits | Garbled Circuits

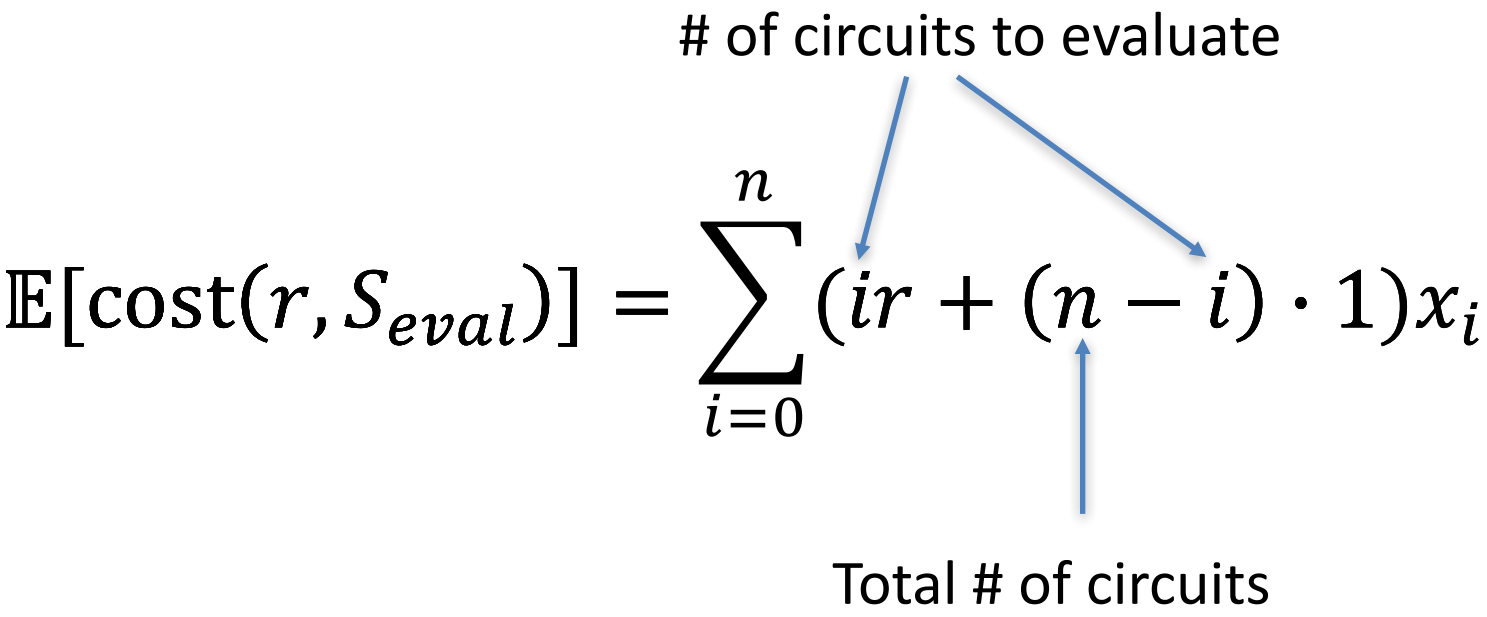
Failure: $S_{gen} = S_{eval}$

So could I

Chk 0 | Eval 1 | Chk 0 | Eval 1 | Eval 1



Expected Cost of SingleCut



n	Total number of circuits
x_i	Probability of evaluating i circuits

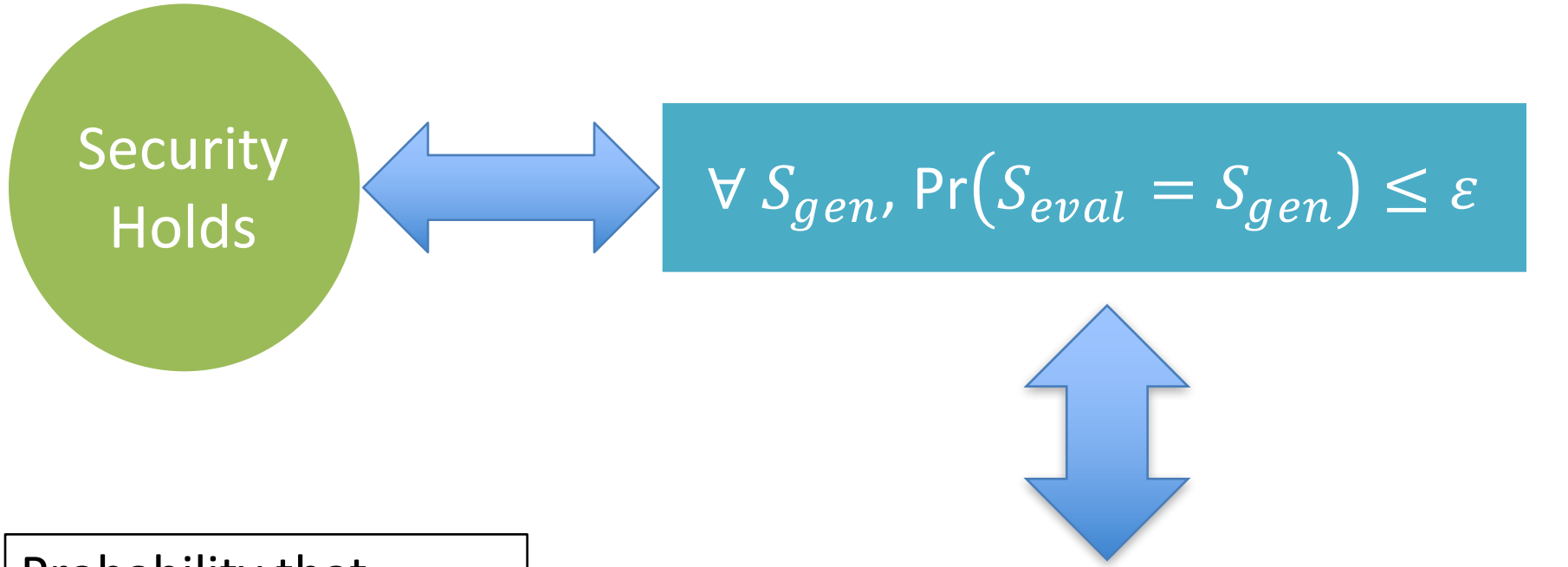
Constraints on x_i (because it's a probability distribution)

$$x_i \geq 0$$

$$\sum_{i=0}^{n} x_i = 1$$

n	Total number of circuits
x_i	Probability of evaluating i circuits

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



Probability that evaluator picks any SPECIFIC strategy a is bounded by ε.

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

$$x_i \leq \binom{n}{i} \cdot \varepsilon$$

Each pure strategy can be picked with probability at most ε.

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



Unit Cost:	n	$r+n-1$	$2r+n-2$	...	nr
Units	$\binom{n}{0}$	$\binom{n}{1}$	$\binom{n}{2}$		$\binom{n}{n}$

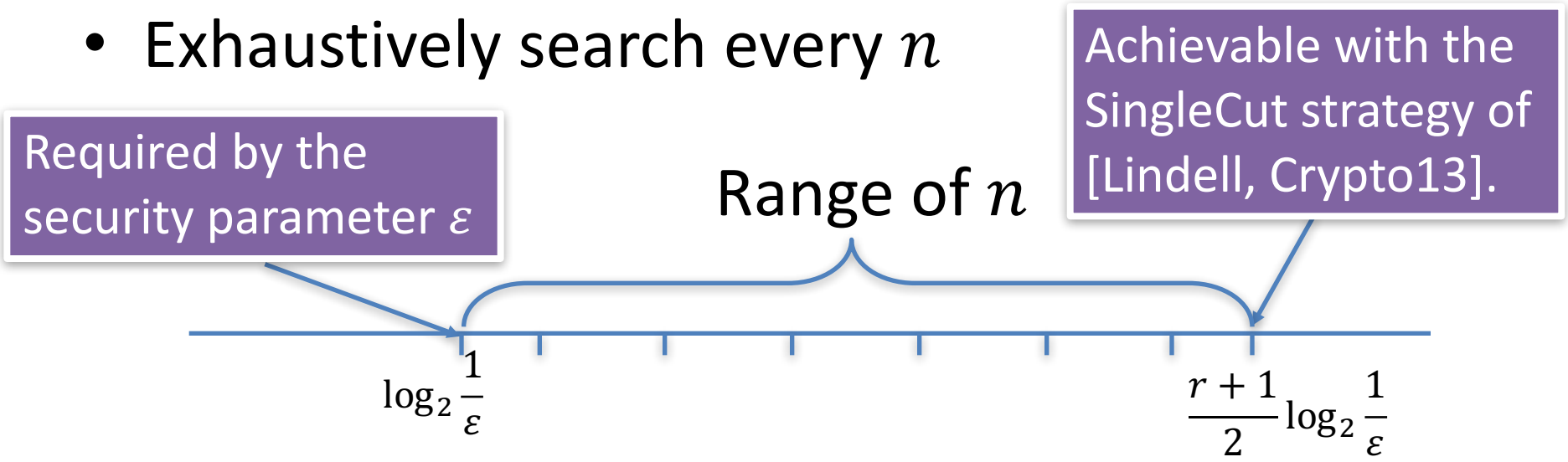
Capacity:
$1/\varepsilon$ units



A greedy algorithm solves it in linear time.

Find the Best n

- Exhaustively search every n



  to find the one with minimal cost.

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

Sample SingleCut Strategy for AES

Classical Strategy

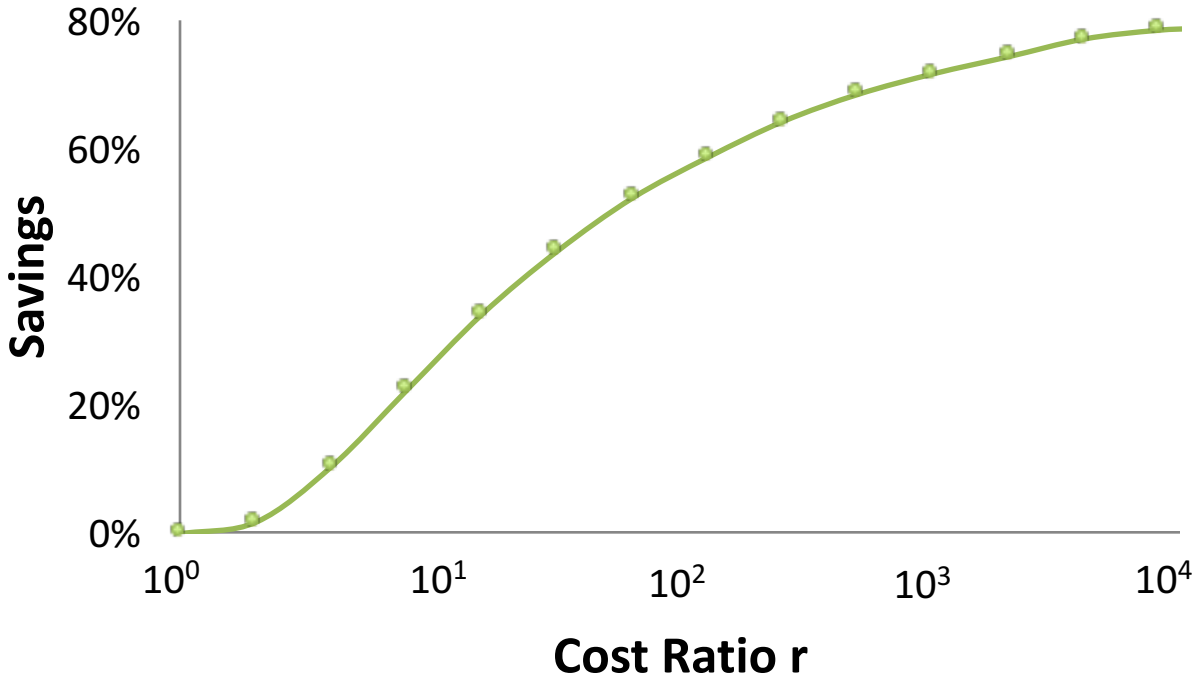
$n = 40$	
i	x_i as %
0	$9.09 \cdot 10^{-11}$
…	…
19	$11.9 \cdot 10^{0}$
20	$12.5 \cdot 10^{0}$
…	…
40	$9.09 \cdot 10^{-11}$

Our technique

$n = 2267$	
i	x_i as %
0	$9.09 \cdot 10^{-11}$
1	$2.06 \cdot 10^{-7}$
2	$2.34 \cdot 10^{-4}$
3	$1.77 \cdot 10^{-1}$
4	$99.8 \cdot 10^{0}$
Save 77.5% b/w	

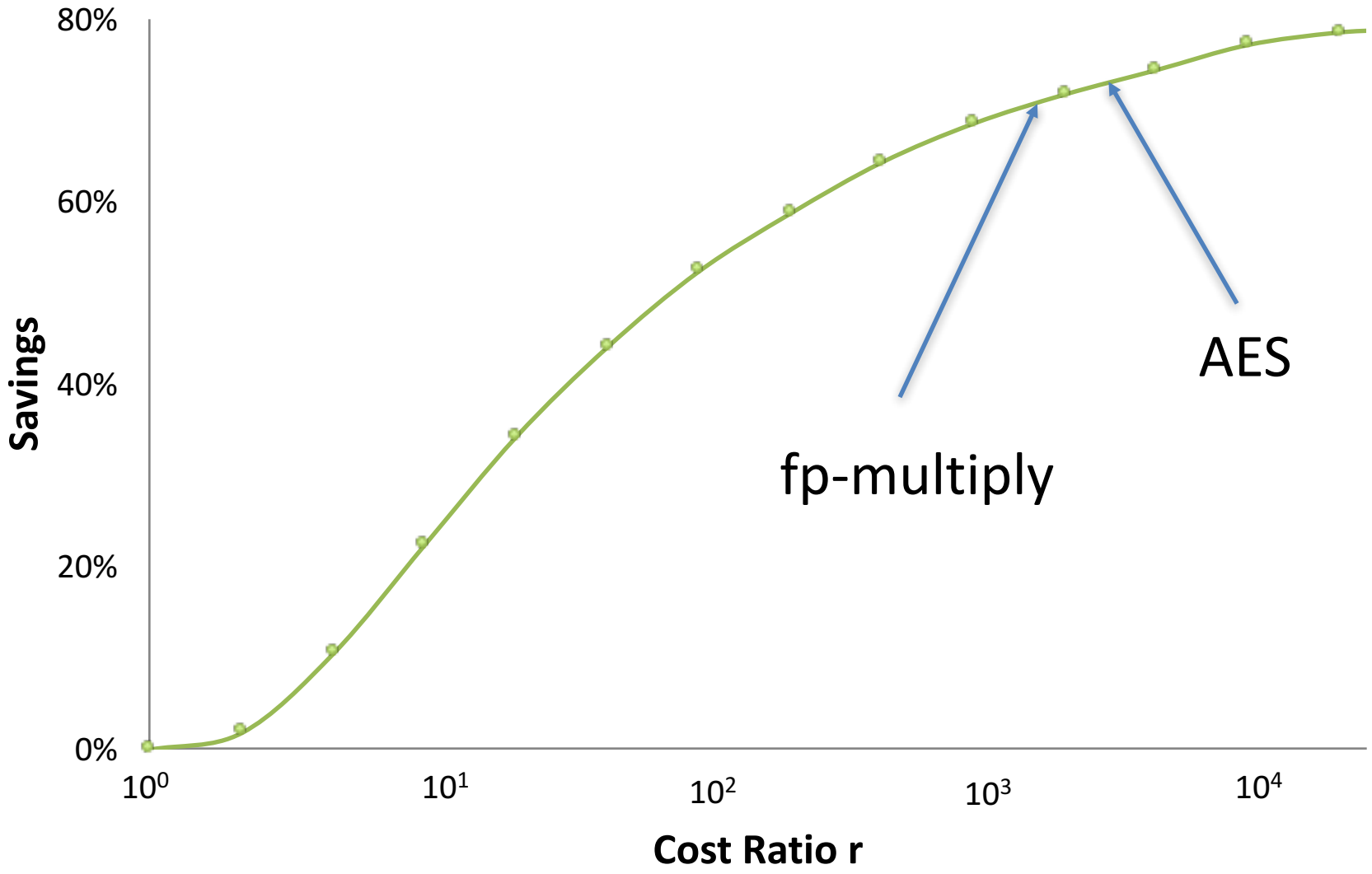
Bandwidth cost ratio: $r = 4533$ For AES

Improvements on SingleCut



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

Improvements on SingleCut



Formulation for MajorityCut

Minimize:

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

Subject to:

$$x_i \geq 0$$

$$\sum_{i=0}^{n} x_i = 1$$

$$\sum_{i=b}^{\min(n,2b)} x_i \cdot \binom{n-b}{i-b} / \binom{n}{i} \leq \varepsilon$$

See the paper for details.

Sample MajorityCut Strategy

Classical Strategy

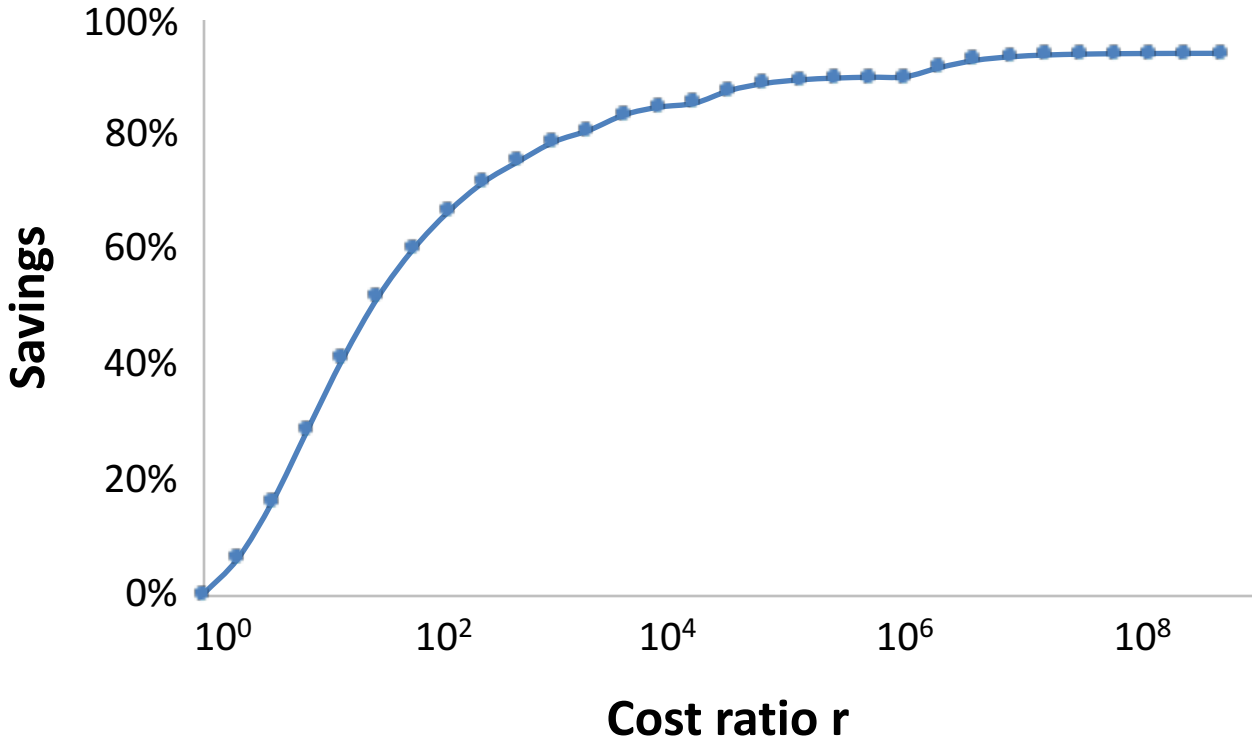
$n = 124$	
i	x_i as %
43	100

Our technique

$n = 175$			
i	x_i as %	i	x_i as %
7	$1 \cdot 10^{-4}$	17	1.23
9	$9 \cdot 10^{-4}$	19	5.36
11	$7 \cdot 10^{-3}$	21	20.9
13	$4.54 \cdot 10^{-2}$	23	72.2
15	0.25		
Save 26.6% time			

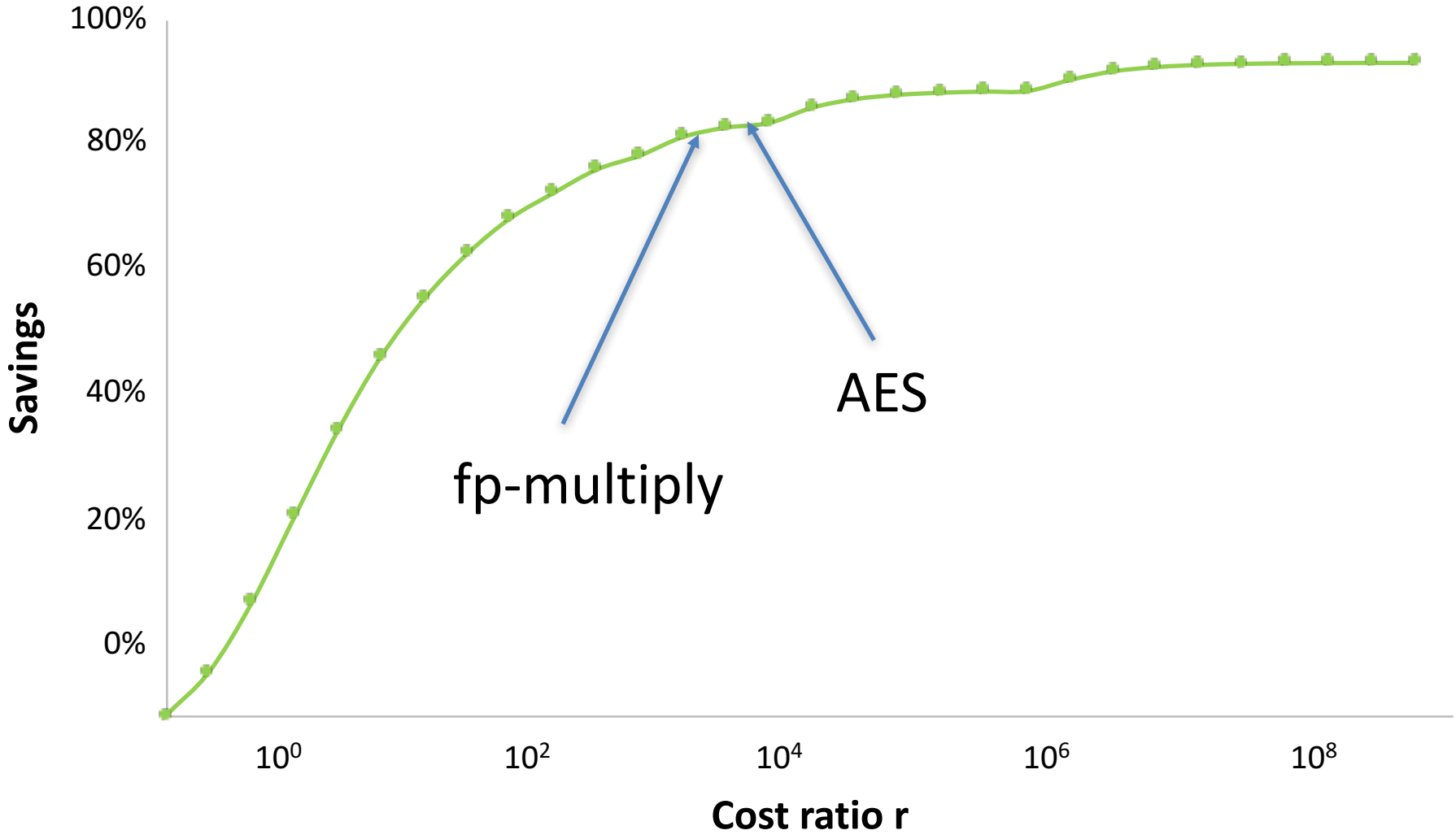
Time cost ratio: $r = 10$

Improvements on MajorityCut

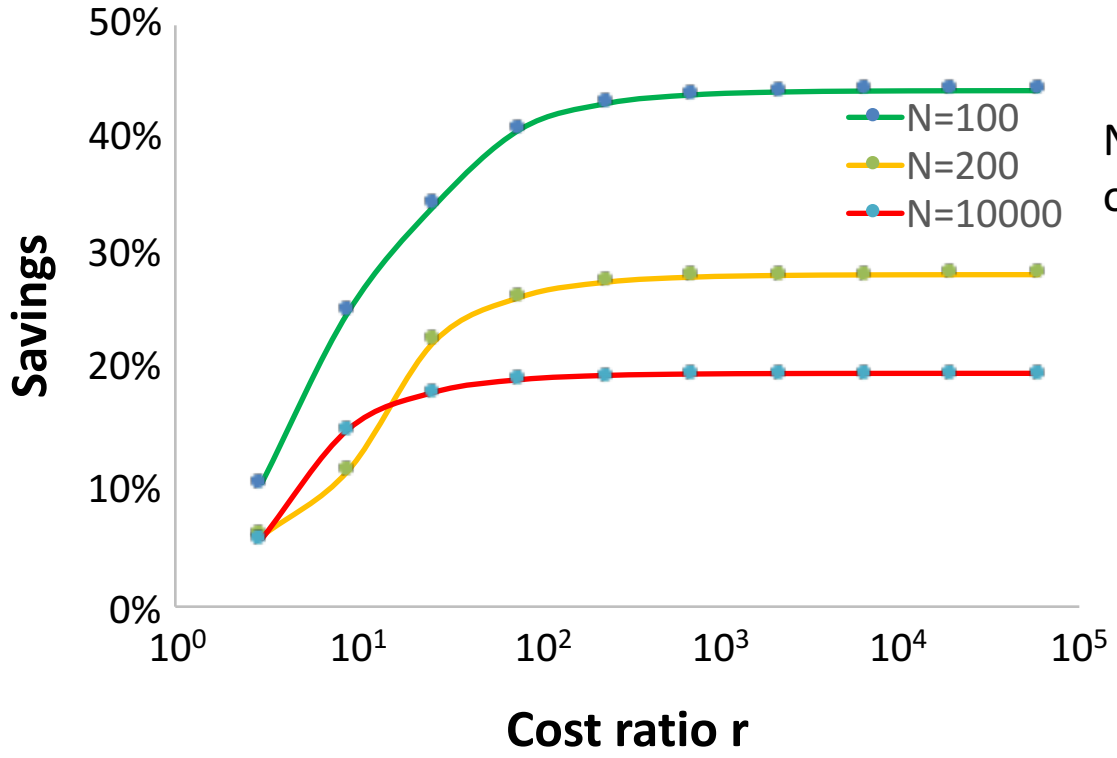


$$\text{Savings} = 1 - \frac{\text{cost}_{\text{this work}}}{\text{cost}_{\text{best prior work}}}$$

Improvements on MajorityCut



Improvements on BatchedCut



N is the size of the circuit.

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

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