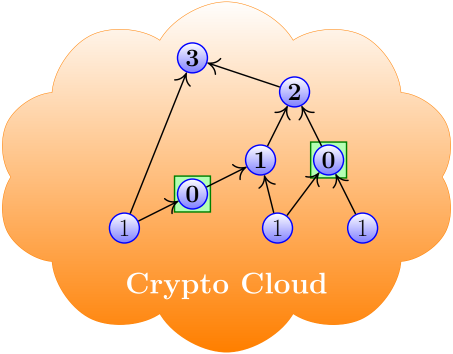
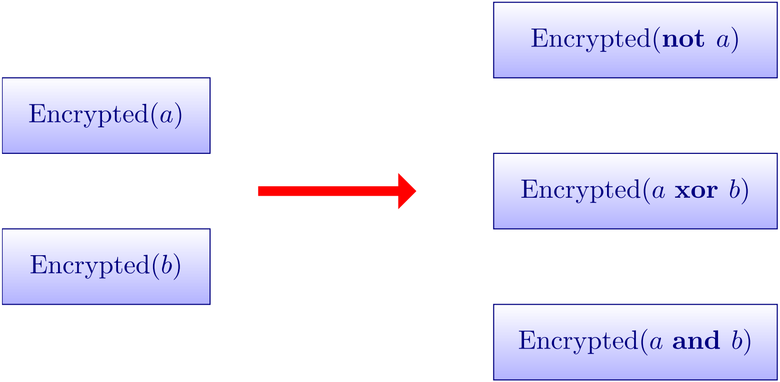
# Optimization of Bootstrapping in Circuits

Fabrice Benhamouda — IBM Research, USA
Tancrède Lepoint — SRI International
Claire Mathieu — École Normale Supérieure
Hang Zhou — Max Planck Institute

SODA'17



# Motivation: Fully Homomorphic Encryption



# Motivation: Noise Level

Encrypted($a$) — $\ell_a$

Encrypted($b$) — $\ell_b$

$\Longrightarrow$

Encrypted(**not** $a$) — $\ell_a$

Encrypted($a$ **xor** $b$) — $\max(\ell_a, \ell_b)$

Encrypted($a$ **and** $b$) — $1 + \max(\ell_a, \ell_b)$

Valid for decryption:

noise level within some parameter $L$ ($L \approx 17$ in practice)

# Motivation: Bootstrap Operations


Goal: Minimize the number of bootstrap operations

# Bootstrap Problem

Input:

- a directed acyclic graph $G = (V, E)$ with two kinds of vertices:

  $\ell = \max(\cdot, \cdot)$

  $\ell = 1 + \max(\cdot, \cdot)$

- an integer parameter $L$

Output:

- a subset $S \subseteq V$ of minimum cardinality such that bootstrapping $S$ ensures $\ell \leq L$ at every vertex

# Previous Results

- Greedy approaches with approximation ratio $\Omega(|V|)$
  [Gentry Halevi 2011; Gentry Halevi Smart 2012]
- Heuristic method
  [Lepoint Paillier 2013]
- Polynomial time algorithm for $L = 1$ and NP-hardness for $L \geq 2$
  [Paindavoine Vialla 2015]

# Our Results

### Approximation

Polynomial-time $L$-approximation algorithm ($L \geq 1$)

Idea: linear program and new rounding scheme

### Inapproximability

UG-hard to compute an $(L - \epsilon)$-approximation ($L \geq 2$)

Idea: reduction to the DAG vertex deletion problem [Svensson 2013]

# Our Results

### Approximation

Polynomial-time $L$-approximation algorithm ($L \geq 1$)

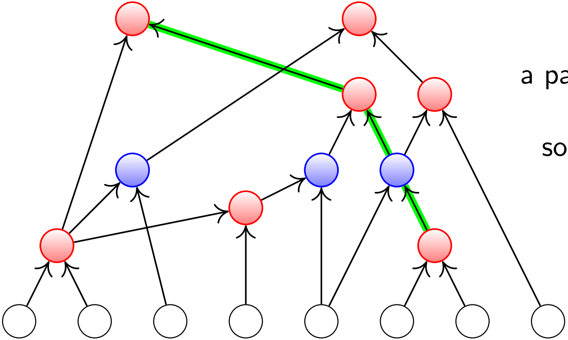
Idea: linear program and new rounding scheme

### Inapproximability

UG-hard to compute an $(L - \epsilon)$-approximation ($L \geq 2$)

Idea: reduction to the DAG vertex deletion problem [Svensson 2013]

# Preliminary Observation



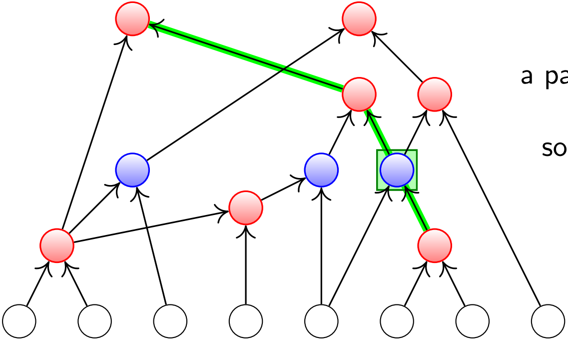
$L = 2$

a path containing 3 red vertices

$\Downarrow$

some vertex is bootstrapped

# Preliminary Observation



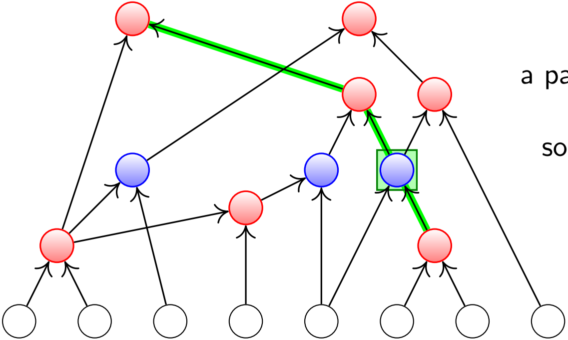
$L = 2$

a path containing 3 red vertices

$\Downarrow$

some vertex is bootstrapped

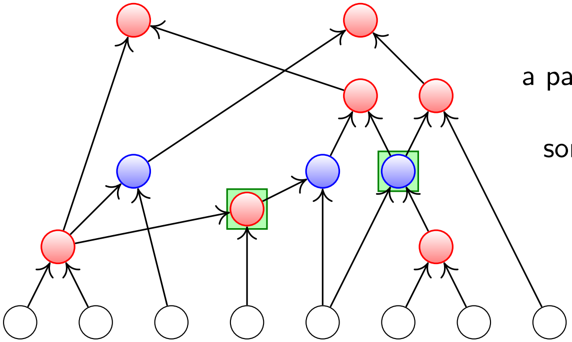
# Preliminary Observation



$L = 2$
a path containing 3 red vertices
$\Downarrow$
some vertex is bootstrapped

Interesting path: containing $L + 1$ red vertices

# Preliminary Observation



$L = 2$
a path containing 3 red vertices
$\Downarrow$
some vertex is bootstrapped

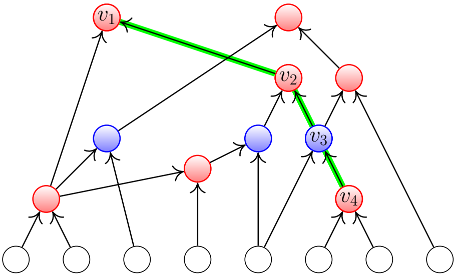
Interesting path: containing $L + 1$ red vertices

**Observation**

bootstrap solution $\iff$ every interesting path has a bootstrapped vertex

# Linear Program Relaxation

$$x_v = \begin{cases} 1 & \text{if } v \text{ is bootstrapped} \\ 0 & \text{otherwise} \end{cases}$$




constraint: $x_{v_1} + x_{v_2} + x_{v_3} + x_{v_4} \geq 1$

$$\begin{aligned} \min \quad & \sum_{v \in V} x_v \\ \text{s.t.} \quad & \sum_{v \in p} x_v \geq 1 && \forall \text{ interesting path } p \\ & 0 \leq x_v \leq 1 && \forall\, v \in V \end{aligned}$$

# Rounding Algorithm

Definition:

- length of a path: sum of $x_v$ along the path
- $f_{v,i}$ : minimum length of a path that ends at $v$ and contains $i$ red vertices
- Interval $A_{v,i} := [f_{v,i}\ ,\ f_{v,i} + x_v]$

## Randomized Rounding

1. Pick $t \in [0, 1]$ uniformly at random
2. For every vertex $v$, bootstrap $v$ if $t \in A_{v,i}$ for some $i \in \{1, \ldots, L\}$.

# Correctness

Need to show:
Every interesting path $v_1, \ldots, v_k$ contains a bootstrapped vertex.

Define $i_j := \#$ red vertices among $v_1, \ldots, v_j$.

**Claim**

The union of $A_{v_j, i_j}$ covers the $[0, 1]$-interval.

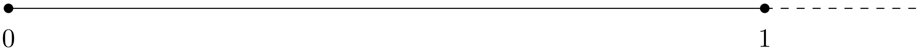
# Correctness

**Claim**

The union of $A_{v_j,i_j}$ covers the $[0,1]$-interval.

Proof:

1. $A_{v_1,i_1}$ starts at 0;
2. every pair of consecutive intervals $A_{v_j,i_j}$ and $A_{v_{j+1},i_{j+1}}$ intersect;
3. $A_{v_k,i_k}$ covers 1.



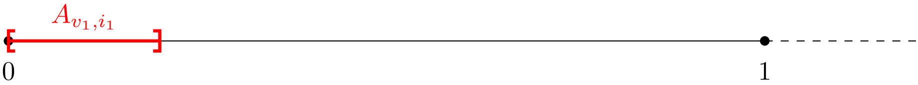
# Correctness

**Claim**

The union of $A_{v_j,i_j}$ covers the $[0,1]$-interval.

Proof:

1. $A_{v_1,i_1}$ starts at 0;
2. every pair of consecutive intervals $A_{v_j,i_j}$ and $A_{v_{j+1},i_{j+1}}$ intersect;
3. $A_{v_k,i_k}$ covers 1.



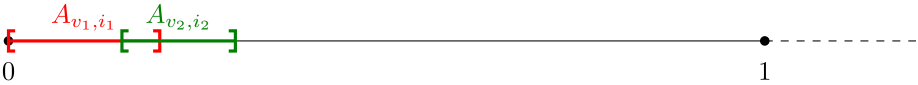
# Correctness

**Claim**

The union of $A_{v_j,i_j}$ covers the $[0,1]$-interval.

Proof:

1. $A_{v_1,i_1}$ starts at 0;
2. every pair of consecutive intervals $A_{v_j,i_j}$ and $A_{v_{j+1},i_{j+1}}$ intersect;
3. $A_{v_k,i_k}$ covers 1.




$f_{v_{j+1},i_{j+1}} \leq f_{v_j,i_j} + x_{v_j}$ by definition of $f$

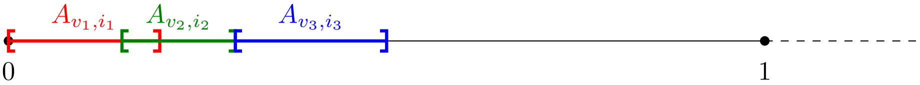
# Correctness

**Claim**

The union of $A_{v_j,i_j}$ covers the $[0,1]$-interval.

Proof:

1. $A_{v_1,i_1}$ starts at 0;
2. every pair of consecutive intervals $A_{v_j,i_j}$ and $A_{v_{j+1},i_{j+1}}$ intersect;
3. $A_{v_k,i_k}$ covers 1.




$f_{v_{j+1},i_{j+1}} \leq f_{v_j,i_j} + x_{v_j}$ by definition of $f$

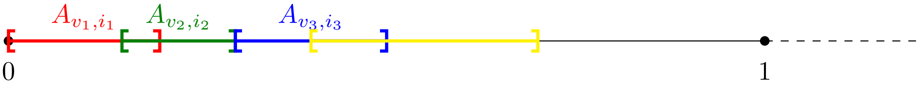
# Correctness

**Claim**

The union of $A_{v_j,i_j}$ covers the $[0,1]$-interval.

Proof:

1. $A_{v_1,i_1}$ starts at 0;
2. every pair of consecutive intervals $A_{v_j,i_j}$ and $A_{v_{j+1},i_{j+1}}$ intersect;
3. $A_{v_k,i_k}$ covers 1.




$f_{v_{j+1},i_{j+1}} \leq f_{v_j,i_j} + x_{v_j}$ by definition of $f$

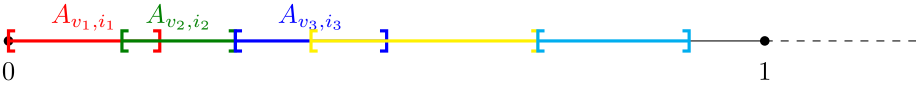
# Correctness

**Claim**

The union of $A_{v_j,i_j}$ covers the $[0,1]$-interval.

Proof:

1. $A_{v_1,i_1}$ starts at 0;
2. every pair of consecutive intervals $A_{v_j,i_j}$ and $A_{v_{j+1},i_{j+1}}$ intersect;
3. $A_{v_k,i_k}$ covers 1.




$f_{v_{j+1},i_{j+1}} \leq f_{v_j,i_j} + x_{v_j}$ by definition of $f$

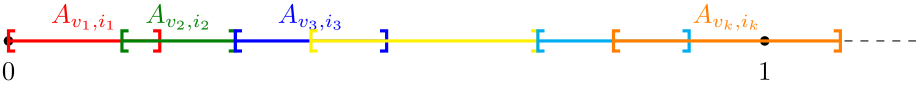
# Correctness

## Claim

The union of $A_{v_j,i_j}$ covers the $[0,1]$-interval.

Proof:

1. $A_{v_1,i_1}$ starts at 0;
2. every pair of consecutive intervals $A_{v_j,i_j}$ and $A_{v_{j+1},i_{j+1}}$ intersect;
3. $A_{v_k,i_k}$ covers 1.




$i_k = L + 1 \implies f_{v_k,i_k} \geq 1$ by definition of $f$ and LP constraints

# $L$-Approximation

A vertex $v$ is bootstrapped if $t \in A_{v,i}$ for some $i \in \{1, \ldots, L\}$.

$\mathbb{P}[v \text{ is bootstapped}] \leq L \cdot x_v$.

Expected number of bootstrapped vertices:

$$\sum_{v \in V} L \cdot x_v \leq L \cdot \mathrm{OPT}.$$

# Conclusion

**Approximation**

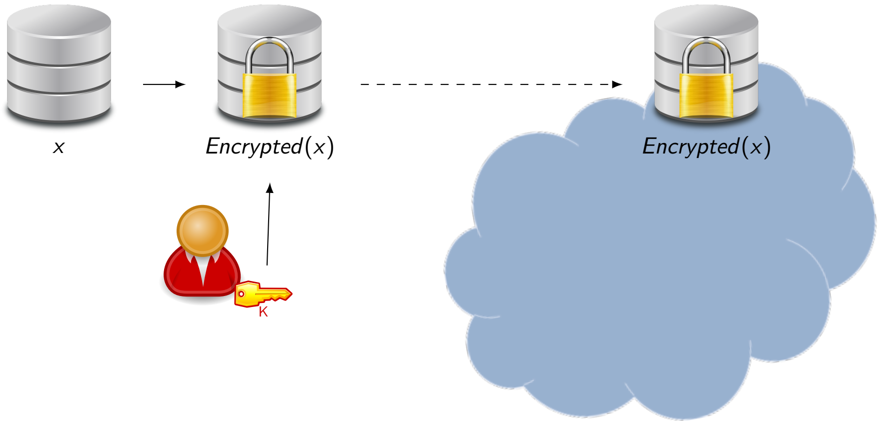
Polynomial-time $L$-approximation algorithm ($L \geq 1$)

**Inapproximability**

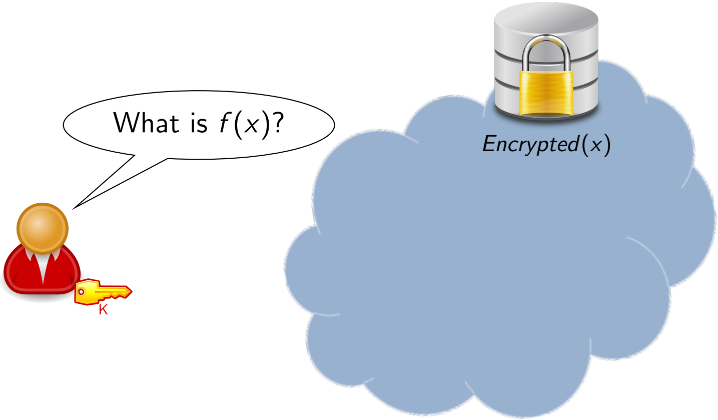
UG-hard to compute an $(L - \epsilon)$-approximation ($L \geq 2$)

Thank you!

# Motivation: Classical Encryption

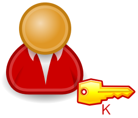


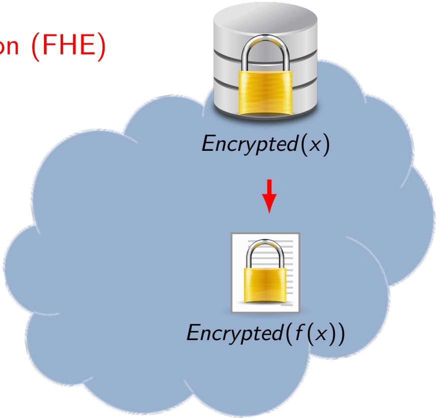
# Motivation: Classical Encryption



# Motivation: Fully Homomorphic Encryption

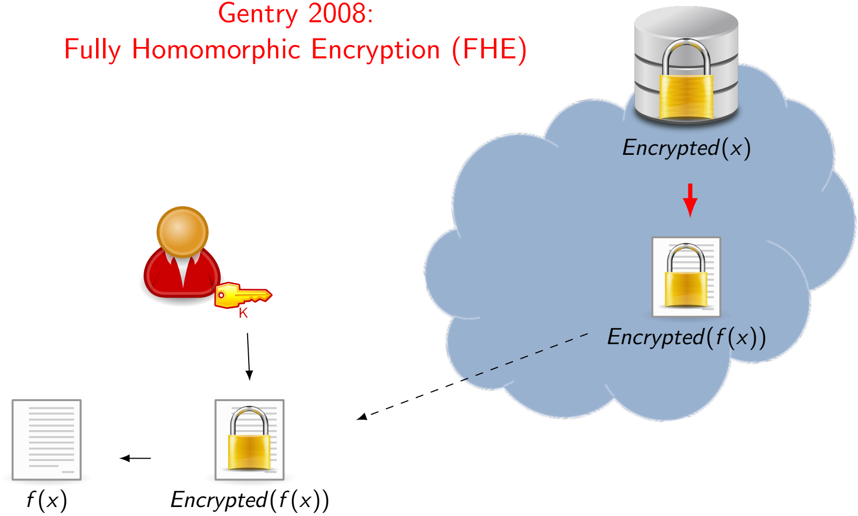
Gentry 2008:
Fully Homomorphic Encryption (FHE)



# Motivation: Fully Homomorphic Encryption

Gentry 2008:
Fully Homomorphic Encryption (FHE)



# Derandomization

$\{f_{v,i}\}_{v,i} \cup \{f_{v,i} + x_v\}_{v,i}$ contains $2|V| \cdot L$ values.

$[0,1]$ interval is decomposed into $O(|V| \cdot L)$ sub-intervals.

## Deterministic Rounding

1. For each sub-interval, pick any $t$ and perform the previous rounding;
2. Return the best solution found.

# Inapproximability

DAG Vertex Deletion (DVD) problem (Svensson 2013):

Input:

- a directed acyclic graph $G = (V, E)$
- an integer parameter $L$

Output:

- a subset $S \subseteq V$ of minimum cardinality such that $G \setminus S$ contains no path of $L$ vertices.

**Inapproximability for DVD [Svensson]**

NP-hard to compute an $(L - \epsilon)$-approximation for the DVD problem ($L \geq 2$), assuming the Unique Games Conjecture

# Inapproximability

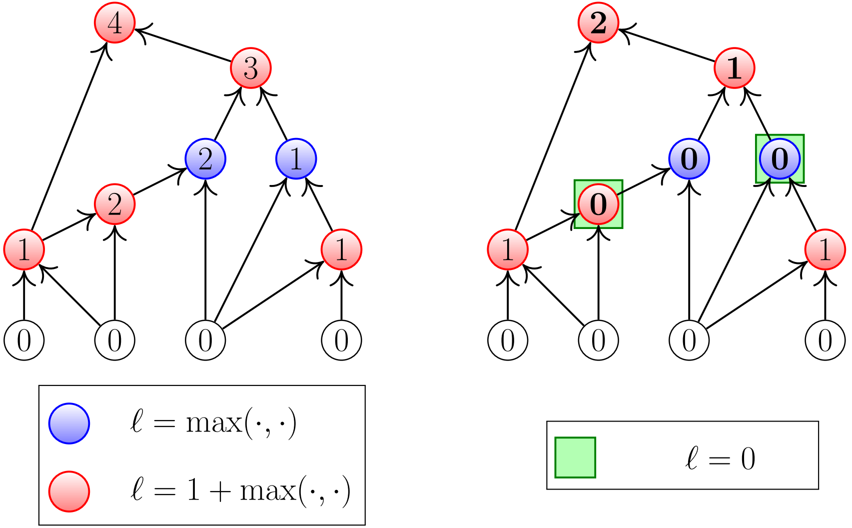
Reduction from DVD to the bootstrap problem:

- DVD instance $\Longleftrightarrow$ bootstrap instance with red and white vertices
- Technical issues: indegree in the DVD problem is not bounded; indegree in the bootstrap problem is bounded by 2

**Inapproximability for bootstrap problem**

NP-hard to compute an $(L - \epsilon)$-approximation for the bootstrap problem $(L \geq 2)$, assuming the Unique Games Conjecture

# Example




$L = 2$