The Cost of Adaptivity in Security Games on Graphs

Chethan Kamath, **Karen Klein**, Krzysztof Pietrzak, Michael Walter

TEL AVIV UNIVERSITY אוניברסיטת תל אביב

ETH zürich

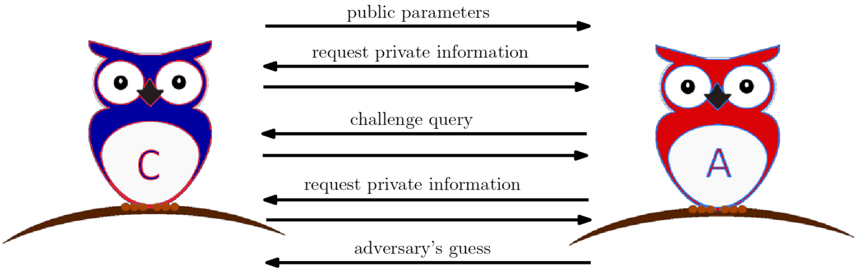
IST AUSTRIA Institute of Science and Technology

ZAMA

November 11, 2021

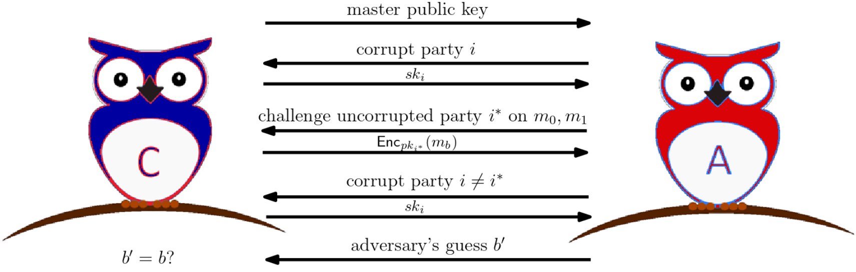
Table of Contents

Introduction: Game-based Security



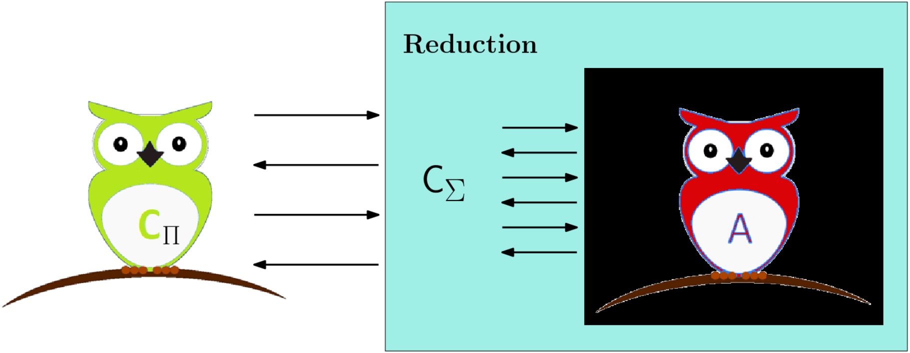
Introduction: Game-based Security

Identity-based Encryption



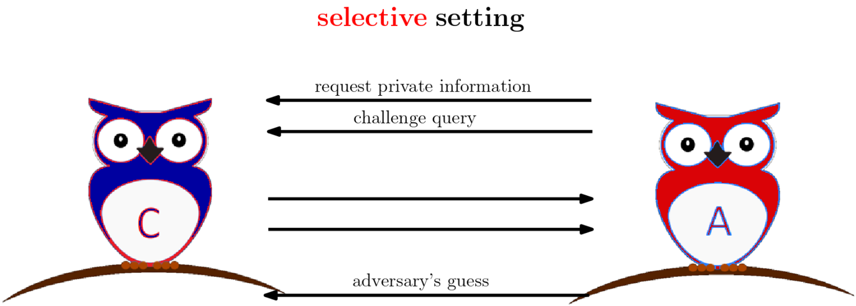
Introduction: Security Proof by Reduction

To prove **security of a scheme Σ**, relate it to some **hard problem Π**



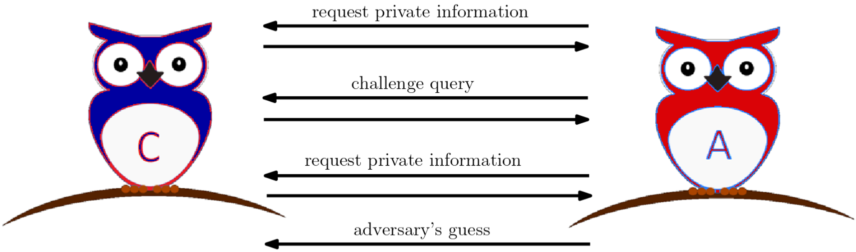
A **breaks Σ** with advantage $\epsilon \Rightarrow$ R **breaks Π** with advantage ϵ/loss

Introduction: Selective versus Adaptive Security



Introduction: Selective versus Adaptive Security

adaptive setting



This paper: Lower bounds on security loss against adaptive adversaries

Our Results

This paper: Lower bounds on security loss against adaptive adversaries

Consider certain multi-round games that capture several existing constructions where the adversary queries edges of a graph:

- **Generalized selective decryption (GSD)**:
  nodes = keys, edges = encryptions
- **TreeKEM construction of continuous group key agreement**:
  nodes = keys, sources = users, sinks = group keys, edges = encryptions
- **GGM84 construction of a prefix-constrained PRFs**:
  nodes = seeds, edges = PRG evaluations
- **Proxy re-encryption (PRE)**:
  nodes = keys, edges = re-encryption keys

Our Results

Application	Underlying Graph	Lower Bound	Reduction	Upper Bound
GSD	Path P_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [FJP15]
	Binary In-Tree B_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [Pan07]
	Tree	$N^{\Omega(\log(N))}$	Straight-line	$N^{O(\log(N))}$ [FJP15]
	Arbitrary DAG	$2^{\Omega(\sqrt{N})}$	Oblivious	$N^{O(N/\log(N))}$ [JKK+17]
TreeKEM	Tree	$M^{\Omega(\log(\log(M)))}$	Straight-line	$Q^{O(\log(M))}$ [KPW+21]
GGM CPRF	Tree	$n^{\Omega(\log(n))}$	Straight-line	$n^{O(\log(n))}$ [FKPR14]
PRE	Path P_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [FKKP19]
	Binary In-Tree B_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [FKKP19]
	Arbitrary DAG	$2^{\Omega(N)}$	Arbitrary	$N^{O(N/\log(N))}$ [FKKP19]

$N = 2^n$ ... size of the graph.
GGM CPRF: n ... input length. TreeKEM: M ... number of users, Q ... number of queries.

Reductions: oblivious $\subseteq$ straight-line $\subseteq$ arbitrary fully black-box

Our Results

Main conceptual idea:

- Introduce **Builder-Pebbler Game**:
  a two-player, multi-stage game
- Pebbler's success probability $\rightarrow$ lower bounds on security loss:
  use oracle separation techniques

Our Results

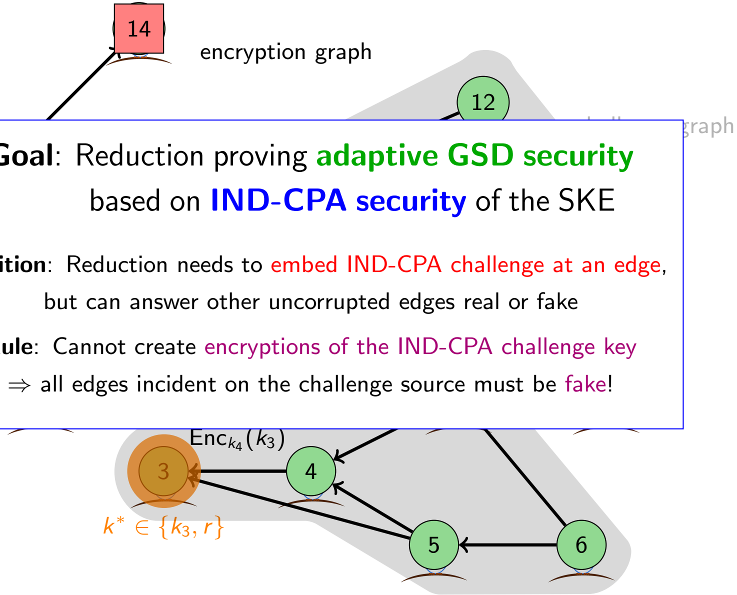
Application	Underlying Graph	Lower Bound	Reduction	Upper Bound
GSD	Path P_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [FJP15]
	Binary In-Tree B_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [Pan07]
	Tree	$N^{\Omega(\log(N))}$	Straight-line	$N^{O(\log(N))}$ [FJP15]
	Arbitrary DAG	$2^{\Omega(\sqrt{N})}$	Oblivious	$N^{O(N/\log(N))}$ [JKK+17]
TreeKEM	Tree	$M^{\Omega(\log(\log(M)))}$	Straight-line	$Q^{O(\log(M))}$ [KPW+21]
GGM CPRF	Tree	$n^{\Omega(\log(n))}$	Straight-line	$n^{O(\log(n))}$ [FKPR14]
PRE	Path P_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [FKKP19]
	Binary In-Tree B_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [FKKP19]
	Arbitrary DAG	$2^{\Omega(N)}$	Arbitrary	$N^{O(N/\log(N))}$ [FKKP19]

$N = 2^n$ ... size of the graph.
GGM CPRF: n ... input length. TreeKEM: M ... number of users, Q ... number of queries.

Reductions: oblivious $\subseteq$ straight-line $\subseteq$ arbitrary fully black-box

Table of Contents

Generalized Selective Decryption (GSD) [Pan07]



Goal: Reduction proving **adaptive GSD security** based on **IND-CPA security** of the SKE

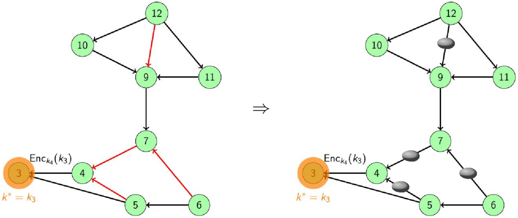
Intuition: Reduction needs to embed IND-CPA challenge at an edge, but can answer other uncorrupted edges real or fake

Rule: Cannot create encryptions of the IND-CPA challenge key $\Rightarrow$ all edges incident on the challenge source must be fake!

Threshold Adversaries

Our (inefficient) adversary:

- Corrupts all nodes outside the challenge graph, outputs 1 if any fake edges outgoing from corrupt nodes
  $\Rightarrow$ challenge key must be embedded in challenge graph
- On the challenge graph: Interprets **fake** edges as **pebbled**



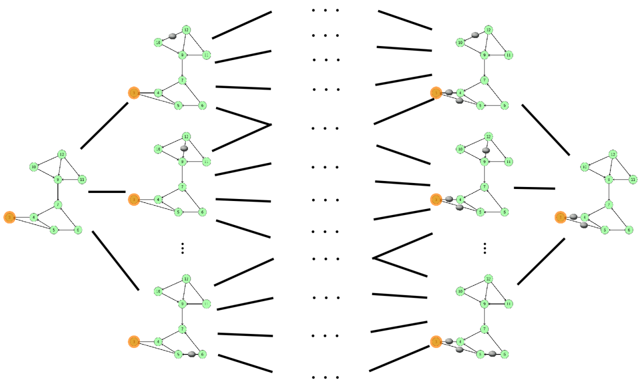
- Outputs 0/1 if final **pebbling configuration** good/bad

Threshold Adversaries

The threshold:

- Consider **reversible edge pebbling**:

  Can place/remove a pebble on an edge iff all edges incident on its source are pebbled.
- Define **good** by a cut in the configuration graph:

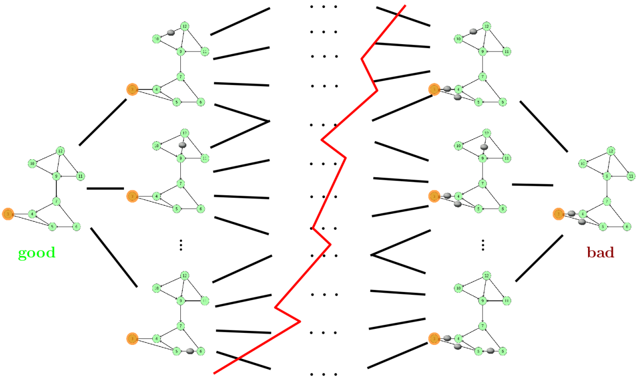


Threshold Adversaries

The threshold:

- Consider **reversible edge pebbling**:

  Can place/remove a pebble on an edge iff all edges incident on its source are pebbled.
- Define **good** by a cut in the configuration graph:

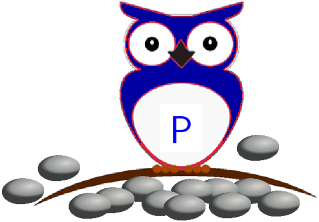


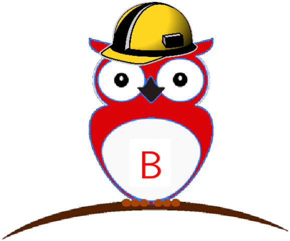
Cut set ... configurations at the border between good and bad

Table of Contents

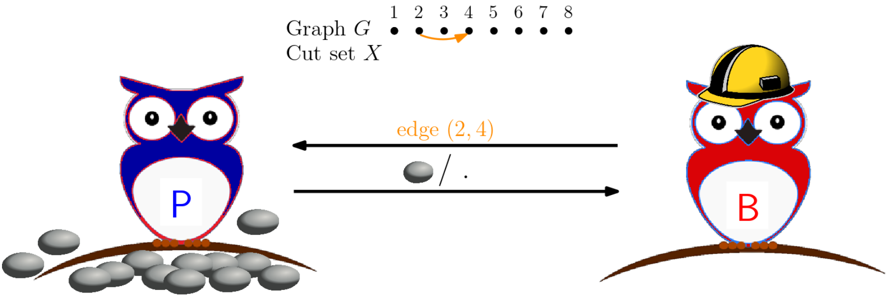
Builder-Pebbler Game

	1	2	3	4	5	6	7	8
Graph G	•	•	•	•	•	•	•	•
Cut set X								

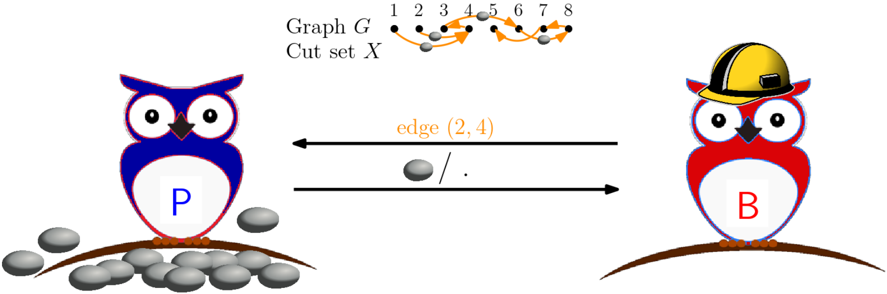




Builder-Pebbler Game

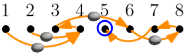


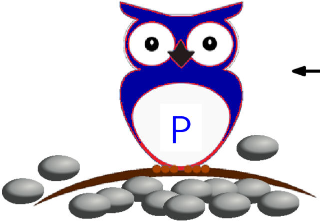
Builder-Pebbler Game



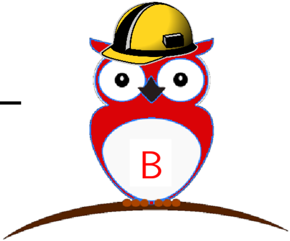
Builder-Pebbler Game

Graph G
Cut set X





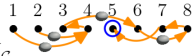
challenge node 5

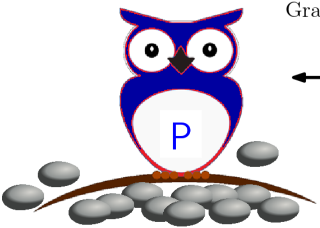


P wins iff $P(G) \in X$

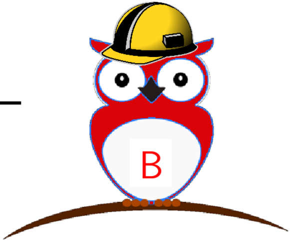
Builder-Pebbler Game

Graph G
Cut set $X_{\mathcal{G}}$
Graph family $\mathcal{G}$



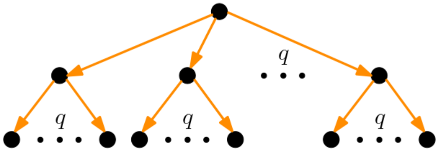


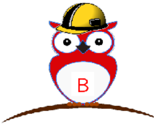
challenge node 5



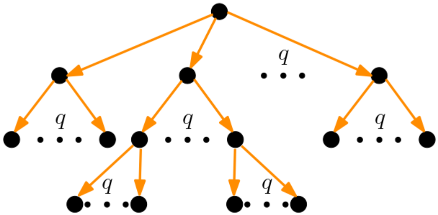
P wins iff $P(G) \in X_{\mathcal{G}}$
$\vee\, G \notin \mathcal{G}$

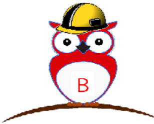
Builder Strategy for Trees



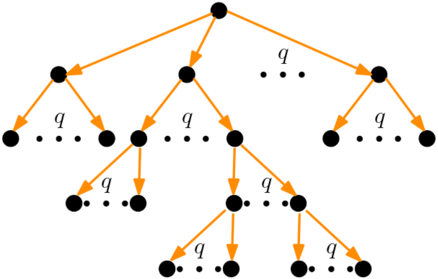


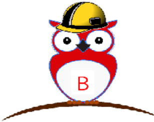
Builder Strategy for Trees



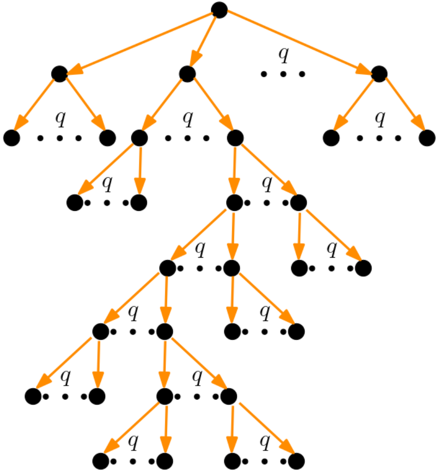


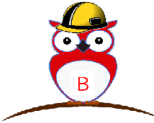
Builder Strategy for Trees



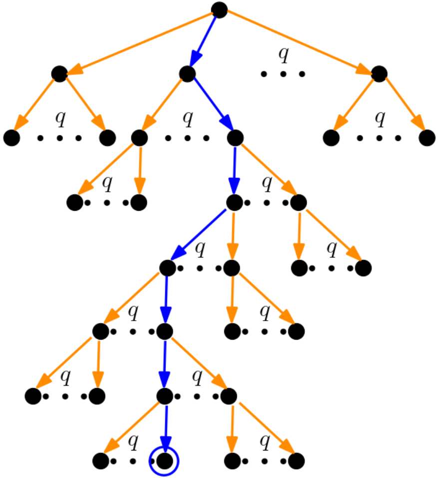


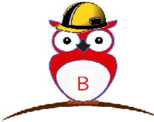
Builder Strategy for Trees





Builder Strategy for Trees



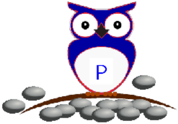


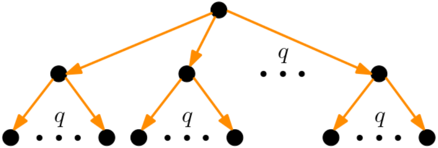
Cut for Trees with Large Outdegree

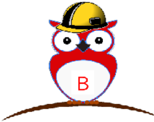
- Challenge graph = path of length n
- Lower bound for reversible edge pebbling on a path:
  Require $\log(n) + 1$ pebbles to pebble last edge
- Define **cut** X: pebble configuration P on the challenge path is good iff it is reachable with $\log(n)$ pebbles

$\Rightarrow$ **Goal of the Pebbler**: Place $\log(n)$ pebbles on the challenge path, but *no* pebbles outgoing from nodes outside the path.

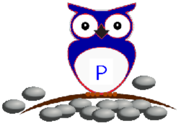
Builder Strategy for Trees

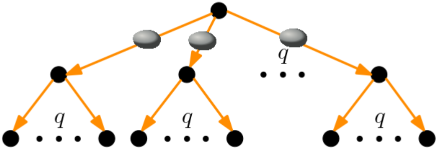


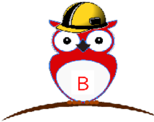




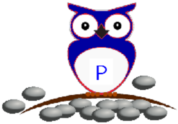
Builder Strategy for Trees

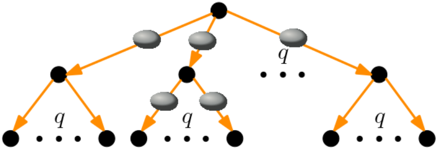


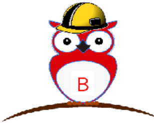




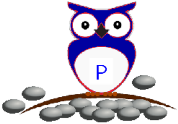
Builder Strategy for Trees

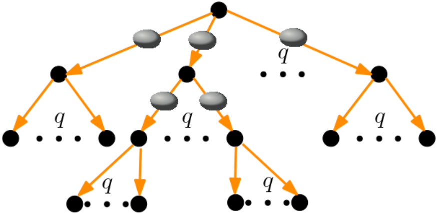


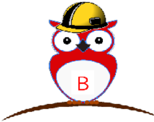




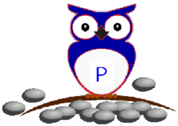
Builder Strategy for Trees

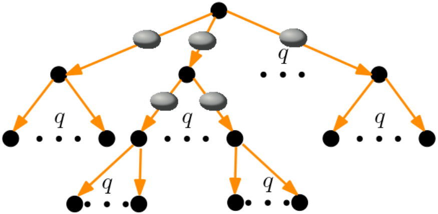


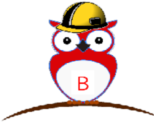




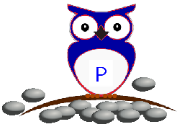
Builder Strategy for Trees

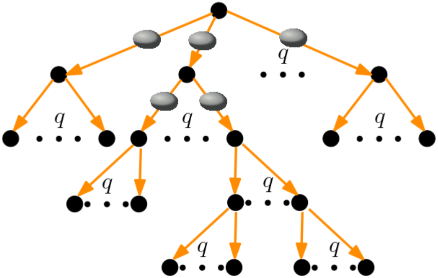


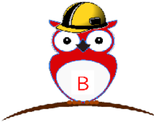




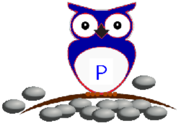
Builder Strategy for Trees

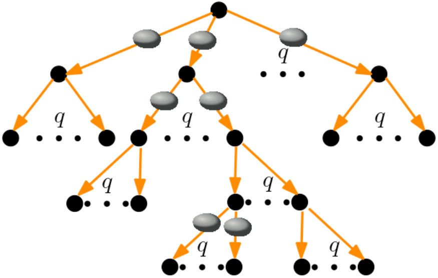


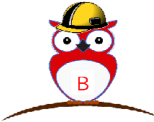




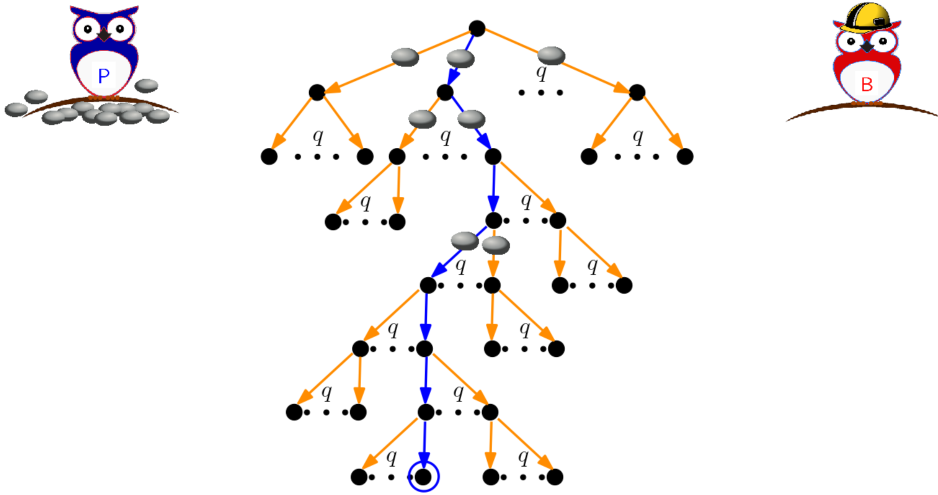
Builder Strategy for Trees



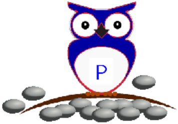


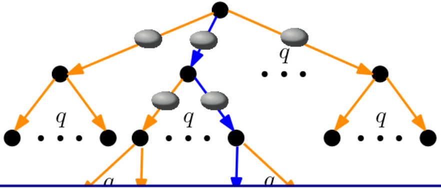


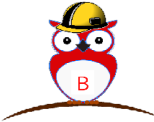
Builder Strategy for Trees



Builder Strategy for Trees







$$\text{Pebbler's success probability} \leq 1/q^{\log(n)-1} \underset{[q=O(n)]}{=} N^{-\Omega(\log(N))}$$

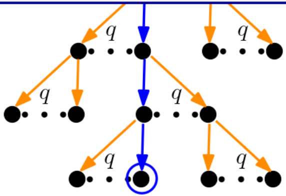


Table of Contents

Lower Bound for GSD

Combinatorial upper bound $\rightarrow$ cryptographic lower bound:

- Construct ideal SKE scheme
- Construct (inefficient) threshold adversary for GSD that simulates the above Builder strategy B, such that:

  $\forall$ straight-line reductions R: $\exists$ Pebbler P against B such that:

$$\text{R has security loss } \leq \Lambda \quad \Rightarrow \quad \text{P has advantage } \geq 1/\Lambda$$

Theorem (GSD on trees, informal)

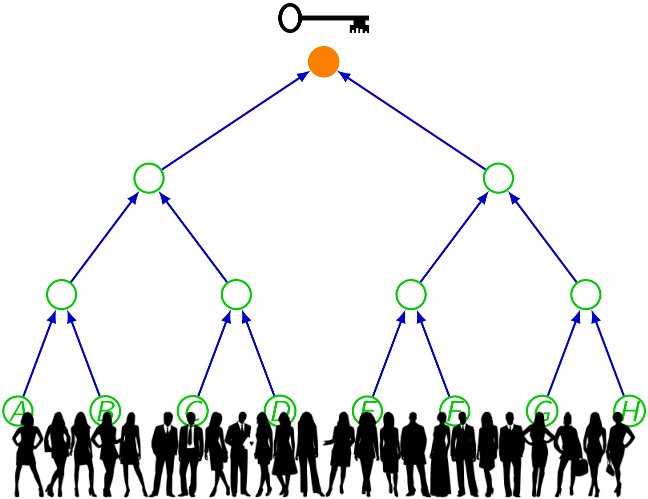
Any **straight-line reduction** *proving security of unrestricted* **adaptive GSD** *based on the IND-CPA security of the underlying SKE scheme* **loses** *at least a* **super-polynomial** *factor ($N^{\Omega(\log(N))}$) in the number of users N.*

Our Results

Application	Underlying Graph	Lower Bound	Reduction	Upper Bound
GSD	Path P_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [FJP15]
	Binary In-Tree B_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [Pan07]
	Tree	$N^{\Omega(\log(N))}$	Straight-line	$N^{O(\log(N))}$ [FJP15]
	Arbitrary DAG	$2^{\Omega(\sqrt{N})}$	Oblivious	$N^{O(N/\log(N))}$ [JKK+17]
TreeKEM	Tree	$M^{\Omega(\log(\log(M)))}$	Straight-line	$Q^{O(\log(M))}$ [KPW+21]
GGM CPRF	Tree	$n^{\Omega(\log(n))}$	Straight-line	$n^{O(\log(n))}$ [FKPR14]
PRE	Path P_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [FKKP19]
	Binary In-Tree B_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [FKKP19]
	Arbitrary DAG	$2^{\Omega(N)}$	Arbitrary	$N^{O(N/\log(N))}$ [FKKP19]

$N = 2^n$ ... size of the graph.
GGM CPRF: n ... input length. TreeKEM: M ... number of users, Q ... number of queries.

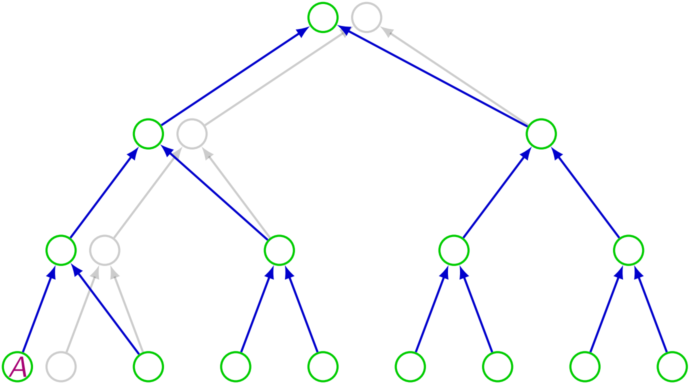
Continuous Group Key Agreement: TreeKEM [BBR18]



TreeKEM: Update

Alice updates:

- choose fresh keys (via hash chain, as in TreeKEM)
- remove old keys



Lower Bound for TreeKEM

- Game is quite similar to **public-key GSD**
- Construct adversary that embeds tree structure as above (depth $\log(M)$, M group size)

  Cruicial: Relay server is not trusted!

Theorem (TreeKEM, informal)

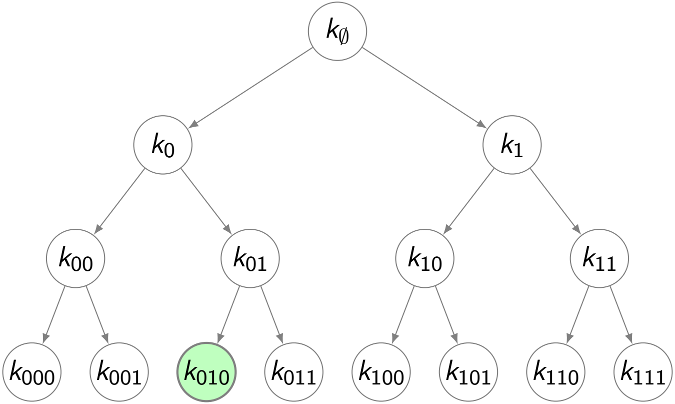
Any **straight-line reduction** *proving* **adaptive CGKA security for TreeKEM** *based on the IND-CPA security of the underlying PKE scheme* **loses** *a* **super-polynomial** *factor* $(M^{\Omega(\log\log(M))})$ *in the group size* M.

Our Results

Application	Underlying Graph	Lower Bound	Reduction	Upper Bound
GSD	Path P_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [FJP15]
	Binary In-Tree B_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [Pan07]
	Tree	$N^{\Omega(\log(N))}$	Straight-line	$N^{O(\log(N))}$ [FJP15]
	Arbitrary DAG	$2^{\Omega(\sqrt{N})}$	Oblivious	$N^{O(N/\log(N))}$ [JKK+17]
TreeKEM	Tree	$M^{\Omega(\log(\log(M)))}$	Straight-line	$Q^{O(\log(M))}$ [KPW+21]
GGM CPRF	Tree	$n^{\Omega(\log(n))}$	Straight-line	$n^{O(\log(n))}$ [FKPR14]
PRE	Path P_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [FKKP19]
	Binary In-Tree B_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [FKKP19]
	Arbitrary DAG	$2^{\Omega(N)}$	Arbitrary	$N^{O(N/\log(N))}$ [FKKP19]

$N = 2^n$ ... size of the graph.
GGM CPRF: n ... input length. TreeKEM: M ... number of users, Q ... number of queries.

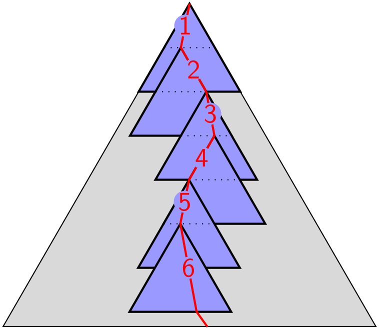
Prefix-constrained PRF: GGM84



$$\mathsf{F}_{GGM}(k, x) = k_x \text{ where } k_\emptyset = k \text{ and } \forall z \in \{0,1\}^* : k_{z\|0} \| k_{z\|1} = \mathsf{PRG}(k_z)$$

Adversary can query constrained keys and evaluations.

Lower Bound for GGM84



Theorem (GGM CPRF, informal)

Any **straight-line reduction** *proving* **adaptive security for the GGM CPRF** *based on the security of the underlying PRG* **loses** *a* **super-polynomial** *factor ($n^{\Omega(\log(n))}$) in the input size n.*

Our Results

Application	Underlying Graph	Lower Bound	Reduction	Upper Bound
GSD	Path P_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [FJP15]
	Binary In-Tree B_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [Pan07]
	Tree	$N^{\Omega(\log(N))}$	Straight-line	$N^{O(\log(N))}$ [FJP15]
	Arbitrary DAG	$2^{\Omega(\sqrt{N})}$	Oblivious	$N^{O(N/\log(N))}$ [JKK+17]
TreeKEM	Tree	$M^{\Omega(\log(\log(M)))}$	Straight-line	$Q^{O(\log(M))}$ [KPW+21]
GGM CPRF	Tree	$n^{\Omega(\log(n))}$	Straight-line	$n^{O(\log(n))}$ [FKPR14]
PRE	Path P_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [FKKP19]
	Binary In-Tree B_N	$N^{\Omega(\log(N))}$	Oblivious	$N^{O(\log(N))}$ [FKKP19]
	Arbitrary DAG	$2^{\Omega(N)}$	Arbitrary	$N^{O(N/\log(N))}$ [FKKP19]

$N = 2^n$ . . . size of the graph.
GGM CPRF: n . . . input length. TreeKEM: M . . . number of users, Q . . . number of queries.

For the other results, see https://eprint.iacr.org/2021/059!

Table of Contents

Conclusion and Open Problems

Initiated study of **lower bounds** on loss in **adaptive security** for certain **multi-round** games on graphs.

- Can we strengthen our lower bounds to hold also for rewinding / non-obliviousness reductions? Or can we use these techniques to overcome our lower bounds?

  PRE on complete DAGs: LB for arbitrary black-box reductions.

- What are other multi-round games captured by the Builder-Pebbler Game?
- Can we use pebbling lower bounds to prove lower bounds on the loss in adaptive security in other settings, i.e. constant-round games (eg. ABE, Garbling)?

  Yao's garbling: Yes [KKPW21], but very different techniques required

THANK YOU FOR YOUR ATTENTION!