Time space tradeoffs for attacks against one-way functions and PRGs

Anindya De[1] Luca Trevisan[2] Madhur Tulsiani[3]

[1]UC Berkeley

[2]UC Berkeley and Stanford University

[3]IAS and Princeton University

What is this talk about?

- Can “brute-force” attacks on cryptographic primitives be improved upon?

What is this talk about?

- Can “brute-force” attacks on cryptographic primitives be improved upon?
  - Recover a key of length k in time less than 2^k.

What is this talk about?

- Can “brute-force” attacks on cryptographic primitives be improved upon?
  - Recover a key of length k in time less than 2^k.
  - In time t, recover key with probability better than $t/2^k$.

What is this talk about?

- Can “brute-force” attacks on cryptographic primitives be improved upon?
  - Recover a key of length k in time less than 2^k.
  - In time t, recover key with probability better than $t/2^k$.
- Are better (non-uniform) attacks possible against:
  - one-way functions?
  - pseudo-random generators?

Definitions of primitives

- $N = 2^n, \quad [N] \cong \{0,1\}^n.$

Definitions of primitives

- $N = 2^n$, $[N] \cong \{0,1\}^n$.
- One-way function: $f : [N] \to [N]$ is (t, ϵ)-one way if for every algorithm A of complexity $\leq t$

$$\Pr_{x \sim \{0,1\}^n} \left[A^f(f(x)) = x' \mid f(x') = f(x) \right] \leq \epsilon$$

Definitions of primitives

- $N = 2^n$, $[N] \cong \{0,1\}^n$.
- One-way function: $f : [N] \to [N]$ is (t, ϵ)-one way if for every algorithm A of complexity $\leq t$

$$\Pr_{x \sim \{0,1\}^n} \left[A^f(f(x)) = x' \;\middle|\; f(x') = f(x) \right] \leq \epsilon$$

- PRG: $G : [N] \to [2N]$ is a (t, ϵ)-secure PRG if for every algorithm A of complexity $\leq t$

$$\left| \Pr_{x \sim [N]} \left[A(G(x)) = 1 \right] - \Pr_{y \sim [2N]} [A(y) = 1] \right| \leq \epsilon$$

Measure of Complexity

- complexity $\neq$ time, as A may compute f^{-1} in $O(\log N)$ time by storing all inverses.

Measure of Complexity

- complexity $\neq$ time, as A may compute f^{-1} in $O(\log N)$ time by storing all inverses.
- complexity $=$ pre-computed advice $+$ running time.

Measure of Complexity

- complexity $\neq$ time, as A may compute f^{-1} in $O(\log N)$ time by storing all inverses.
- complexity $=$ pre-computed advice $+$ running time.
- Can be implemented on a RAM with time and space t.
- Similar to circuit complexity.

Upper bounds

	Primitive	Complexity
[Hellman 80]	Permutation f	$\tilde{O}(\sqrt{N})$

Upper bounds

	Primitive	Complexity
[Hellman 80]	Permutation f	$\tilde{O}(\sqrt{N})$
[Hellman 80]	Random function f (heuristic)	$\tilde{O}(N^{2/3})$

Upper bounds

	Primitive	Complexity
[Hellman 80]	Permutation f	$\tilde{O}(\sqrt{N})$
[Hellman 80]	Random function f (heuristic)	$\tilde{O}(N^{2/3})$
[Fiat-Naor 99]	Any f, all inputs	$\tilde{O}(N^{3/4})$

Upper bounds

	Primitive	Complexity	
[Hellman 80]	Permutation f	$\tilde{O}(\sqrt{N})$	
[Hellman 80]	Random function f (heuristic)	$\tilde{O}(N^{2/3})$	
[Fiat-Naor 99]	Any f, all inputs	$\tilde{O}(N^{3/4})$	
[DTT 10]	Any f, ϵ-fraction of inputs	$\tilde{O}(\sqrt{\epsilon N})$	$\epsilon \leq N^{-1/3}$
		$\tilde{O}(\epsilon^{5/4} N^{3/4})$	$\epsilon \geq N^{-1/3}$

Upper bounds

	Primitive	Complexity
[Hellman 80]	Permutation f	$\tilde{O}(\sqrt{N})$
[Hellman 80]	Random function f (heuristic)	$\tilde{O}(N^{2/3})$
[Fiat-Naor 99]	Any f, all inputs	$\tilde{O}(N^{3/4})$
[DTT 10]	Any f, ϵ-fraction of inputs	$\tilde{O}(\sqrt{\epsilon N})$ $\quad \epsilon \leq N^{-1/3}$ $\tilde{O}(\epsilon^{5/4} N^{3/4})$ $\quad \epsilon \geq N^{-1/3}$
[ACR 97]	PRG $G(x) \stackrel{\text{def}}{=} (f(x), P(x))$	$\tilde{O}(\epsilon^2 N)$

Upper bounds

	Primitive	Complexity
[Hellman 80]	Permutation f	$\tilde{O}(\sqrt{N})$
[Hellman 80]	Random function f (heuristic)	$\tilde{O}(N^{2/3})$
[Fiat-Naor 99]	Any f, all inputs	$\tilde{O}(N^{3/4})$
[DTT 10]	Any f, ϵ-fraction of inputs	$\tilde{O}(\sqrt{\epsilon N})$ $\quad \epsilon \leq N^{-1/3}$ $\tilde{O}(\epsilon^{5/4} N^{3/4})$ $\quad \epsilon \geq N^{-1/3}$
[ACR 97]	PRG $G(x) \stackrel{\text{def}}{=} (f(x), P(x))$	$\tilde{O}(\epsilon^2 N)$
[DTT 10]	Any PRG	$\tilde{O}(\epsilon^2 N)$

Upper bounds

	Primitive	Complexity
[Hellman 80]	Permutation f	$\tilde{O}(\sqrt{N})$
[Hellman 80]	Random function f (heuristic)	$\tilde{O}(N^{2/3})$
[Fiat-Naor 99]	Any f, all inputs	$\tilde{O}(N^{3/4})$
[DTT 10]	Any f, ϵ-fraction of inputs	$\tilde{O}(\sqrt{\epsilon N})$ $\quad \epsilon \leq N^{-1/3}$ $\tilde{O}(\epsilon^{5/4} N^{3/4})$ $\quad \epsilon \geq N^{-1/3}$
[ACR 97]	PRG $G(x) \stackrel{\text{def}}{=} (f(x), P(x))$	$\tilde{O}(\epsilon^2 N)$
[DTT 10]	Any PRG	$\tilde{O}(\epsilon^2 N)$

All above results are actually stated as time-space tradeoffs. Complexity is optimized when $T = S$.

Lower bounds

Better stated in terms of a tradeoff between T and S.

Lower bounds

Better stated in terms of a tradeoff between T and S.

	Primitive	Tradeoff
[Yao 90] [Gennaro-Trevisan 00] [Wee 05]	Permutation f, ϵ-fraction of inputs	$T \cdot S = \tilde{\Omega}(\epsilon N)$ for $T = O(\sqrt{\epsilon N})$

Lower bounds

Better stated in terms of a tradeoff between T and S.

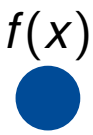
	Primitive	Tradeoff
[Yao 90] [Gennaro-Trevisan 00] [Wee 05]	Permutation f, ϵ-fraction of inputs	$T \cdot S = \tilde{\Omega}(\epsilon N)$ for $T = O(\sqrt{\epsilon N})$
[DTT 10]	Permutation f, ϵ-fraction of inputs	$T \cdot S = \tilde{\Omega}(\epsilon N)$ for any T

Lower bounds

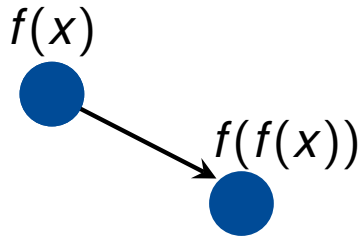
Better stated in terms of a tradeoff between T and S.

	Primitive	Tradeoff
[Yao 90] [Gennaro-Trevisan 00] [Wee 05]	Permutation f, ϵ-fraction of inputs	$T \cdot S = \tilde{\Omega}(\epsilon N)$ for $T = O(\sqrt{\epsilon N})$
[DTT 10]	Permutation f, ϵ-fraction of inputs	$T \cdot S = \tilde{\Omega}(\epsilon N)$ for any T
[DTT 10]	PRG $G \stackrel{\text{def}}{=} (f(x), P(x))$	$T \cdot S = \Omega(\epsilon^2 N)$

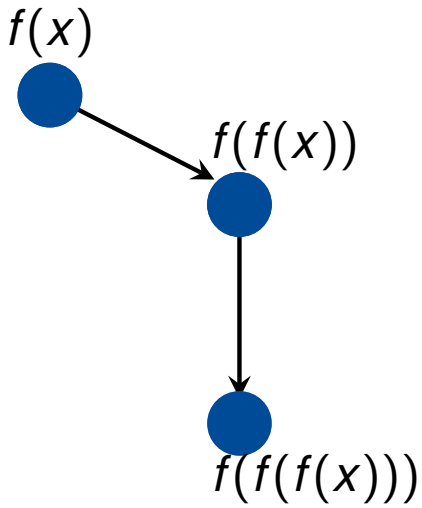
Hellman's approach for permutations



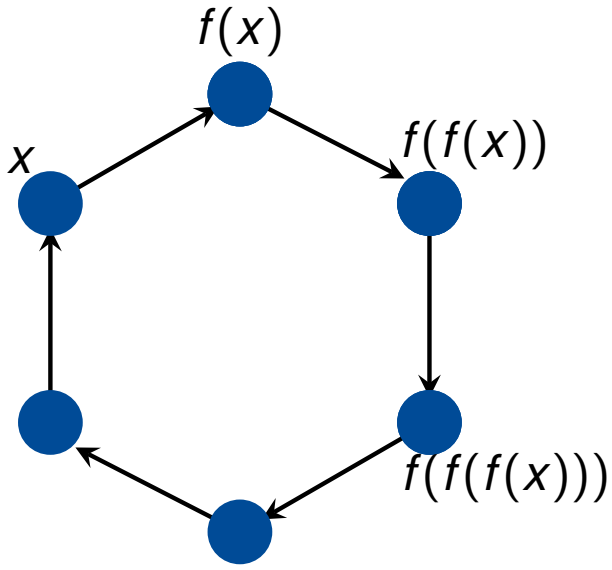
Hellman's approach for permutations



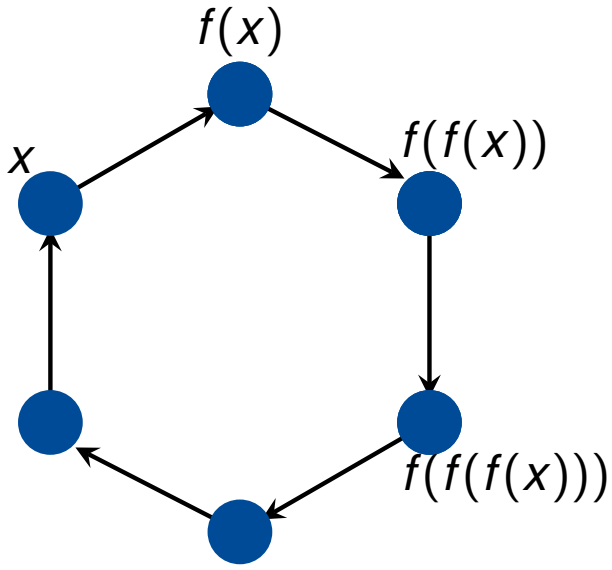
Hellman's approach for permutations

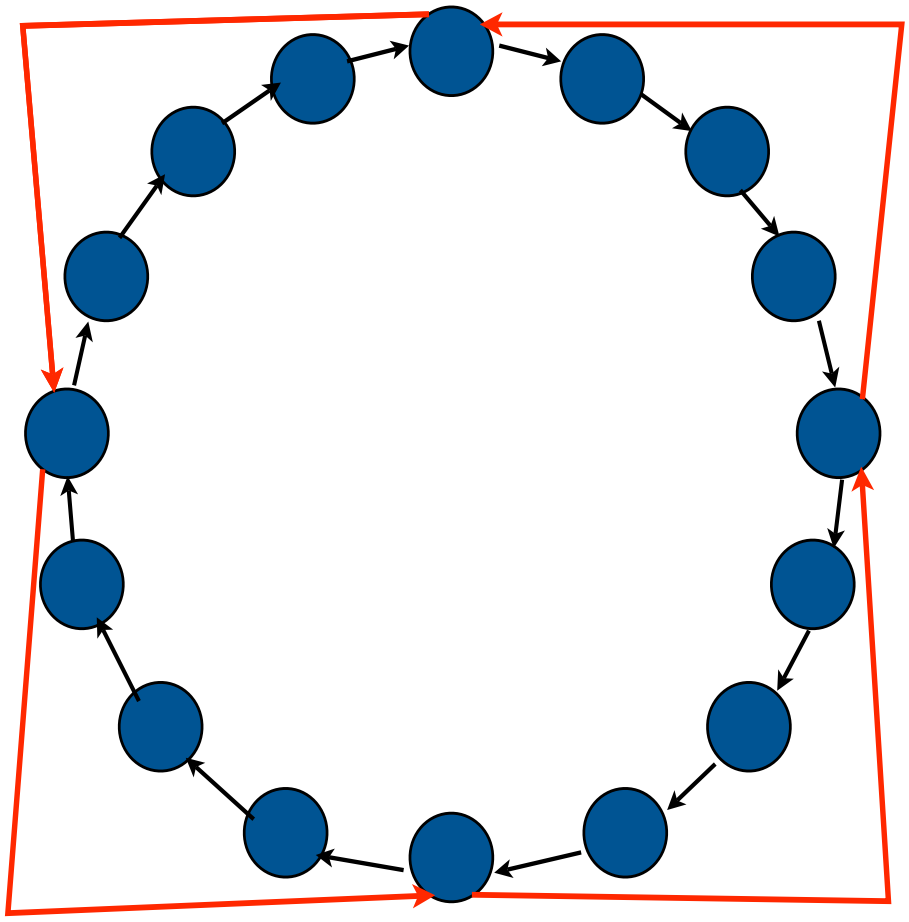


Hellman's approach for permutations

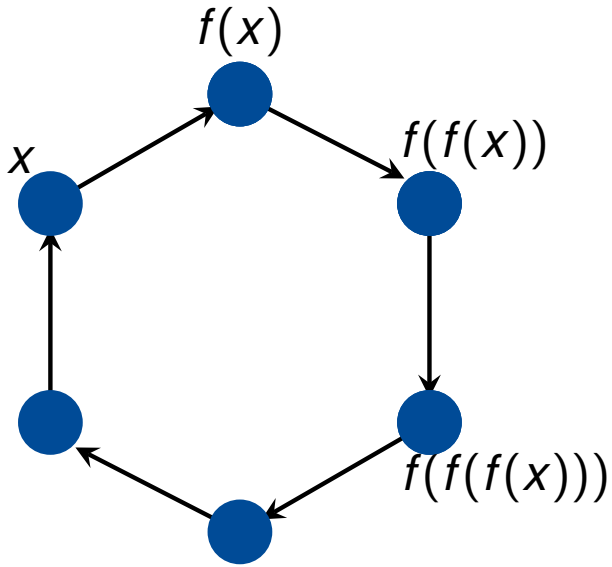


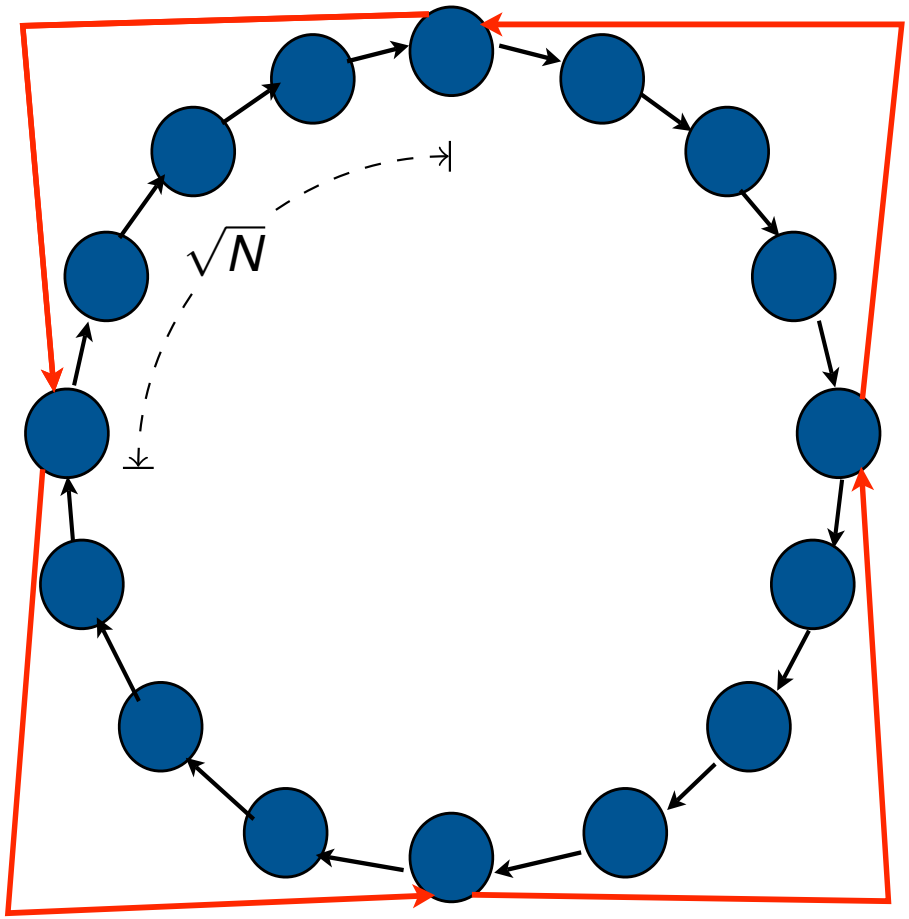
Hellman's approach for permutations



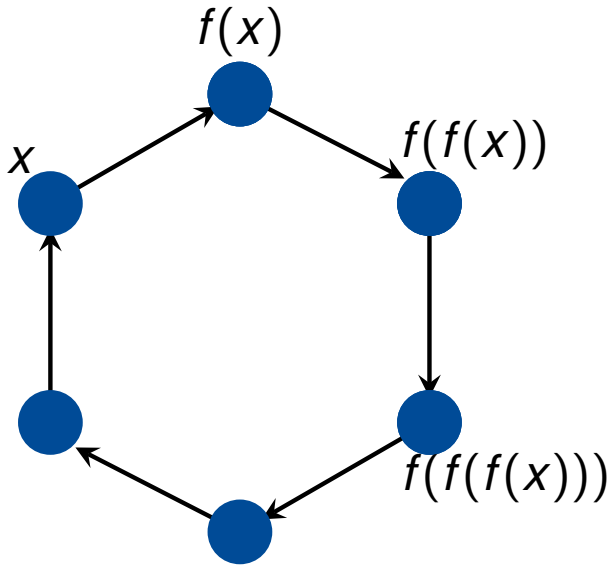


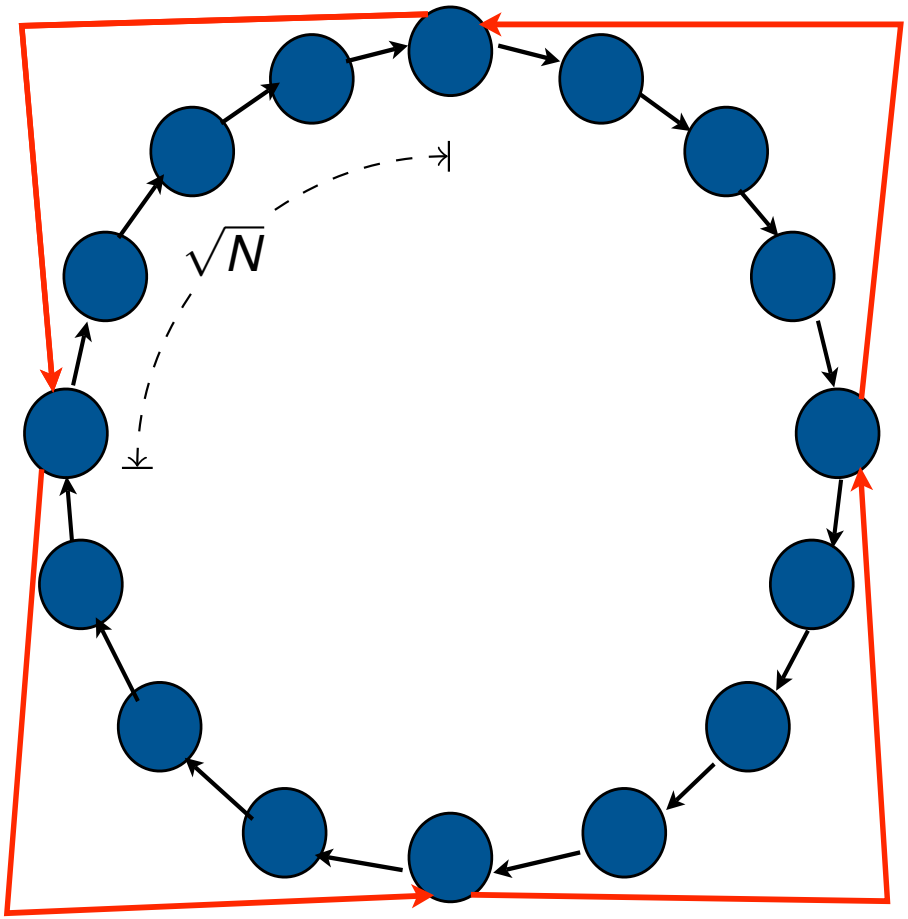
Hellman's approach for permutations





Hellman's approach for permutations





In large cycles, store back-links at distance $\sqrt{N}$. $T, S = O(\sqrt{N})$.

Abstracting the approach for permutations

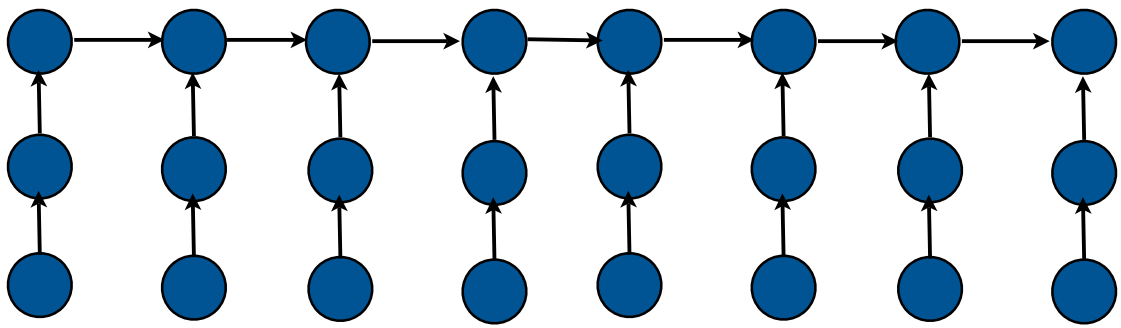
- Cover the graph $(x \to f(x))$ of f by m disjoint paths of length ℓ.

Abstracting the approach for permutations

- Cover the graph $(x \to f(x))$ of f by m disjoint paths of length ℓ.
- Gives algo with $T = \tilde{O}(\ell)$ and $S = \tilde{O}(m)$ (one back-link per path).

Abstracting the approach for permutations

- Cover the graph $(x \to f(x))$ of f by m disjoint paths of length ℓ.
- Gives algo with $T = \tilde{O}(\ell)$ and $S = \tilde{O}(m)$ (one back-link per path).
- Problem: m may have to be very large.



Approach for random functions [Hellman, Fiat-Naor]

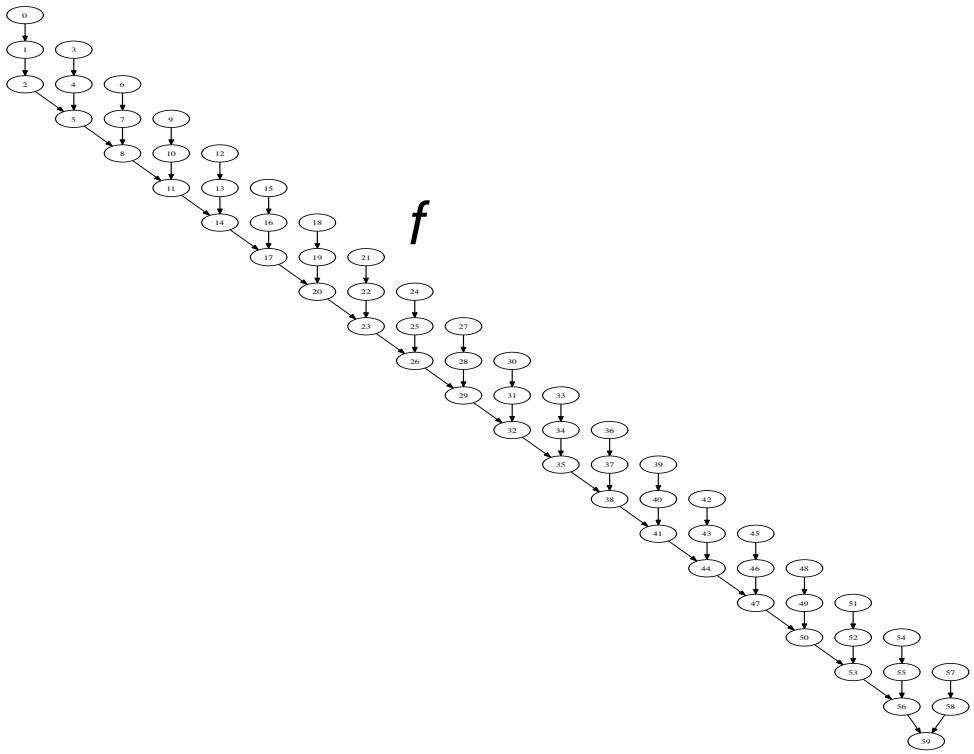
- Collision probability: $\lambda = \mathsf{Pr}_{x,x' \sim [N]} [f(x) = f(x')]$.

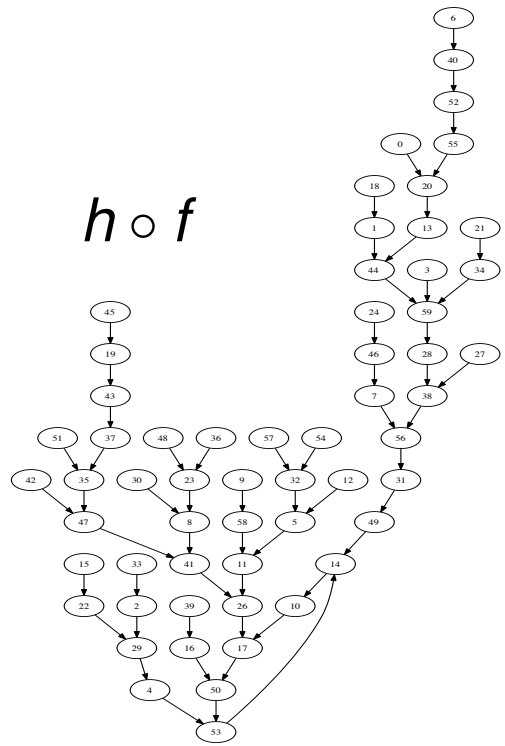
Approach for random functions [Hellman, Fiat-Naor]

- Collision probability: $\lambda = \Pr_{x,x' \sim [N]}[f(x) = f(x')]$.
- If h is a (known) permutation, then inverting $h \circ f$ suffices. If h is random and f has low collision probability, then $h \circ f$ has many long paths.

Approach for random functions [Hellman, Fiat-Naor]

- Collision probability: $\lambda = \Pr_{x,x' \sim [N]}[f(x) = f(x')]$.
- If h is a (known) permutation, then inverting $h \circ f$ suffices. If h is random and f has low collision probability, then $h \circ f$ has many long paths.





Inverting random functions ($\lambda \approx 1/N$)

- For independent random permutations $h_1, \ldots, h_r$, let $g_i = h_i \circ f$.

Inverting random functions ($\lambda \approx 1/N$)

- For independent random permutations $h_1, \ldots, h_r$, let $g_i = h_i \circ f$.
- For each g_i, can find m disjoint paths of length ℓ as long as $m \cdot \ell^2 \cdot \lambda \ll 1$. (each g_i inverts $m \cdot \ell$ elements).

Inverting random functions ($\lambda \approx 1/N$)

- For independent random permutations $h_1, \ldots, h_r$, let $g_i = h_i \circ f$.
- For each g_i, can find m disjoint paths of length ℓ as long as $m \cdot \ell^2 \cdot \lambda \ll 1$. (each g_i inverts $m \cdot \ell$ elements).
- If $g_i's$ behave independently, elements inverted by each of them are independent. Overall $O(m \cdot \ell \cdot r)$ elements inverted.

Inverting random functions ($\lambda \approx 1/N$)

- For independent random permutations $h_1, \ldots, h_r$, let $g_i = h_i \circ f$.
- For each g_i, can find m disjoint paths of length ℓ as long as $m \cdot \ell^2 \cdot \lambda \ll 1$. (each g_i inverts $m \cdot \ell$ elements).
- If $g_i's$ behave independently, elements inverted by each of them are independent. Overall $O(m \cdot \ell \cdot r)$ elements inverted.
- Choose $m, \ell, r = \tilde{O}(N^{1/3})$.

$$\begin{aligned} T &= O(\ell \cdot r) = \tilde{O}(N^{2/3}) \\ S &= O(m \cdot r) = \tilde{O}(N^{2/3}) \end{aligned}$$

Inverting random functions ($\lambda \approx 1/N$)

- For independent random permutations $h_1, \ldots, h_r$, let $g_i = h_i \circ f$.
- For each g_i, can find m disjoint paths of length ℓ as long as $m \cdot \ell^2 \cdot \lambda \ll 1$. (each g_i inverts $m \cdot \ell$ elements).
- If $g_i's$ behave independently, elements inverted by each of them are independent. Overall $O(m \cdot \ell \cdot r)$ elements inverted.
- Choose $m, \ell, r = \tilde{O}(N^{1/3})$.

$$\begin{aligned} T &= O(\ell \cdot r) = \tilde{O}(N^{2/3}) \\ S &= O(m \cdot r) = \tilde{O}(N^{2/3}) \end{aligned}$$

- Problems: Computing $h_1, \ldots, h_r$. Restricted to random f.

Inverting arbitrary functions [Fiat-Naor]

- Store a table of K elements with many pre-images. Collision probability restricted to the remaining inputs is $\approx 1/K$.

Inverting arbitrary functions [Fiat-Naor]

- Store a table of K elements with many pre-images. Collision probability restricted to the remaining inputs is $\approx 1/K$.
- Each h_i only needs to be an ℓ-wise independent hash function. Also, $h_1, \ldots, h_r$ only need to be pairwise independent.

Inverting arbitrary functions [Fiat-Naor]

- Store a table of K elements with many pre-images. Collision probability restricted to the remaining inputs is $\approx 1/K$.
- Each h_i only needs to be an ℓ-wise independent hash function. Also, $h_1, \ldots, h_r$ only need to be pairwise independent.
- Amortize time for one evauation each of $h_1, \ldots, h_r$ to $\tilde{O}(\ell + r)$.

$$T = (\text{time to compute } h_1, \ldots, h_r) \cdot \ell = \tilde{O}(\ell^2 + \ell \cdot r)$$

$$S = \tilde{O}(K + m \cdot r)$$

Inverting arbitrary functions [Fiat-Naor]

- Store a table of K elements with many pre-images. Collision probability restricted to the remaining inputs is $\approx 1/K$.
- Each h_i only needs to be an ℓ-wise independent hash function. Also, $h_1, \ldots, h_r$ only need to be pairwise independent.
- Amortize time for one evauation each of $h_1, \ldots, h_r$ to $\tilde{O}(\ell + r)$.

$$T = (\text{time to compute } h_1, \ldots, h_r) \cdot \ell = \tilde{O}(\ell^2 + \ell \cdot r)$$

$$S = \tilde{O}(K + m \cdot r)$$

- Can again choose m, l such that $m\ell^2\lambda \approx m\ell^2/K \ll 1$. Can get

$$T, S = \tilde{O}(N^{3/4})$$

by taking $K = \tilde{O}(N^{3/4})$, $r = \tilde{O}(N^{1/2})$ and $m, \ell = \tilde{O}(N^{1/4})$.

Inverting f on ϵ-fraction of inputs

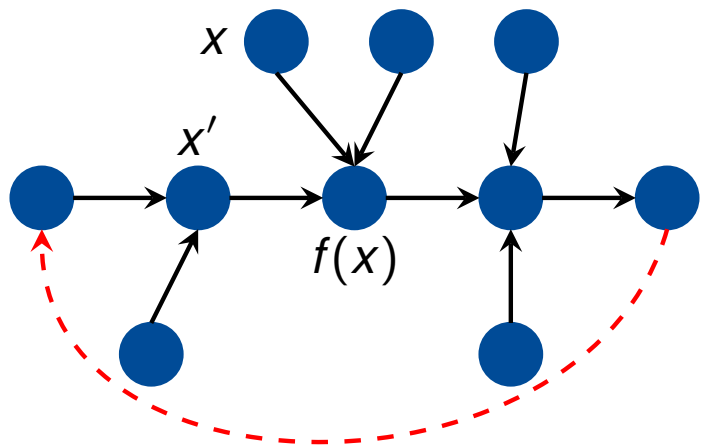
- Directly scaling the Fiat-Naor result would give complexity $(\epsilon N)^{3/4}$ (we claimed $\epsilon^{5/4} N^{3/4}$). Improved analysis using two simple ideas.

Inverting f on ϵ-fraction of inputs

- Directly scaling the Fiat-Naor result would give complexity $(\epsilon N)^{3/4}$ (we claimed $\epsilon^{5/4} N^{3/4}$). Improved analysis using two simple ideas.
- First observation: If a table of size K does not invert f with probability ϵ, then the collision probability for the rest is ϵ/K.

Inverting f on ϵ-fraction of inputs

- Directly scaling the Fiat-Naor result would give complexity $(\epsilon N)^{3/4}$ (we claimed $\epsilon^{5/4} N^{3/4}$). Improved analysis using two simple ideas.
- First observation: If a table of size K does not invert f with probability ϵ, then the collision probability for the rest is ϵ/K.
- Second Observation: The number of elements inverted by a path is not just the path length, but the the sum of indegrees of elements in the path.



Issues in analysis

- Problem: Probablities for inversion of moderately high indegree elements do not add up (with our parameters) in the graphs for $g_1, \ldots, g_r$. Also, these may not be in the table.

Issues in analysis

- Problem: Probablities for inversion of moderately high indegree elements do not add up (with our parameters) in the graphs for $g_1, \ldots, g_r$. Also, these may not be in the table.
  - Analyze these separately using a weaker bound.
  - Either weaker bound suffices or get better control on collision probability.

Issues in analysis

- Problem: Probablities for inversion of moderately high indegree elements do not add up (with our parameters) in the graphs for $g_1, \ldots, g_r$. Also, these may not be in the table.
  - Analyze these separately using a weaker bound.
  - Either weaker bound suffices or get better control on collision probability.

- Problem: Value of r is $O(1)$ for some ranges of ϵ and amortization over evaluations of $h_1, \ldots, h_r$ is not possible.

Issues in analysis

- Problem: Probablities for inversion of moderately high indegree elements do not add up (with our parameters) in the graphs for $g_1, \ldots, g_r$. Also, these may not be in the table.
  - Analyze these separately using a weaker bound.
  - Either weaker bound suffices or get better control on collision probability.

- Problem: Value of r is $O(1)$ for some ranges of ϵ and amortization over evaluations of $h_1, \ldots, h_r$ is not possible.
  - Use better construction based on lossless expanders of Capalbo et al. [CRVW02] and an observation of Seigel [Seigel89].
  - Take $\ell^{o(1)}$ time per evaluation.

Issues in analysis

- Problem: Probablities for inversion of moderately high indegree elements do not add up (with our parameters) in the graphs for $g_1, \ldots, g_r$. Also, these may not be in the table.
  - Analyze these separately using a weaker bound.
  - Either weaker bound suffices or get better control on collision probability.

- Problem: Value of r is $O(1)$ for some ranges of ϵ and amortization over evaluations of $h_1, \ldots, h_r$ is not possible.
  - Use better construction based on lossless expanders of Capalbo et al. [CRVW02] and an observation of Seigel [Seigel89].
  - Take $\ell^{o(1)}$ time per evaluation.

- Final complexity: $T, S = \begin{array}{ll} \tilde{O}(\sqrt{\epsilon N}) & \epsilon \leq N^{-1/3} \\ \tilde{O}(\epsilon^{5/4} N^{3/4}) & \epsilon \geq N^{-1/3} \end{array}$

Lower bound for inverting permutations

- Given A inverting f on ϵ fraction of inputs in time T and space S, want to show $T \cdot S = \Omega(\epsilon N)$.

Lower bound for inverting permutations

- Given A inverting f on ϵ fraction of inputs in time T and space S, want to show $T \cdot S = \Omega(\epsilon N)$.
- Showed by [Yao90] for $\epsilon = 1$ and [GT00], [Wee05] when $T = O(\sqrt{\epsilon N})$.

Lower bound for inverting permutations

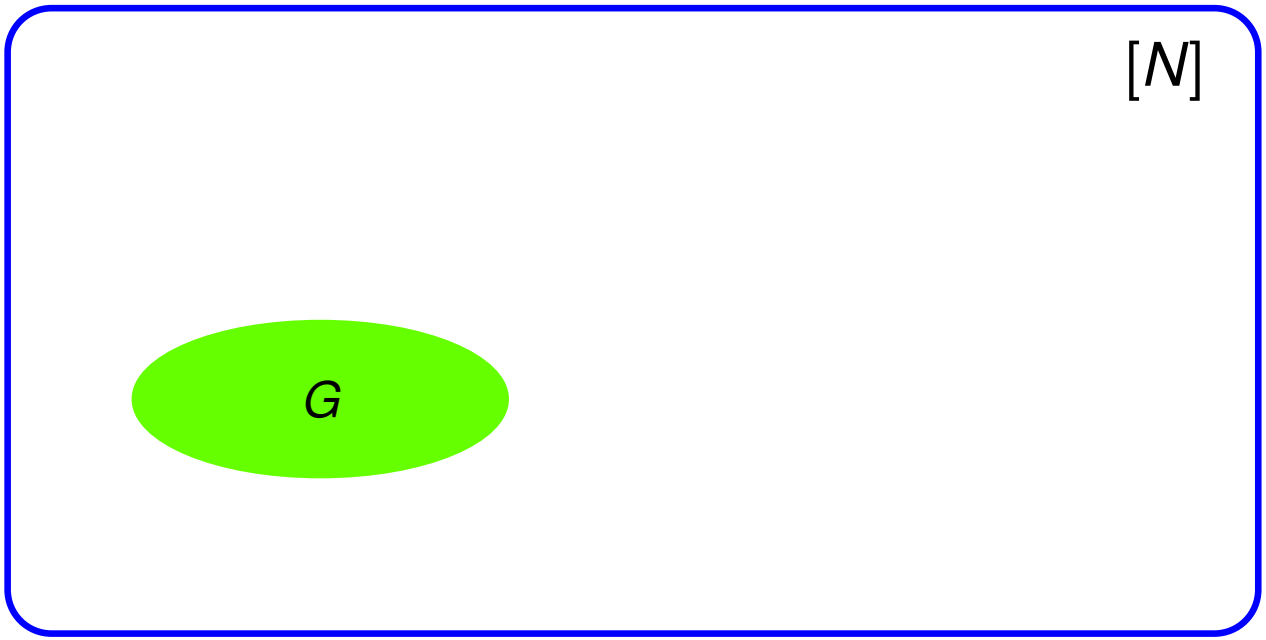
- Given A inverting f on ϵ fraction of inputs in time T and space S, want to show $T \cdot S = \Omega(\epsilon N)$.
- Showed by [Yao90] for $\epsilon = 1$ and [GT00], [Wee05] when $T = O(\sqrt{\epsilon N})$.
- Give a simpler, "randomized" proof that works for all T. Also extends to lower bounds for PRGs.

Lower bound for inverting permutations

- Given A inverting f on ϵ fraction of inputs in time T and space S, want to show $T \cdot S = \Omega(\epsilon N)$.
- Showed by [Yao90] for $\epsilon = 1$ and [GT00], [Wee05] when $T = O(\sqrt{\epsilon N})$.
- Give a simpler, "randomized" proof that works for all T. Also extends to lower bounds for PRGs.
- As in [GT00], show that using A, can encode f with $\approx \log(N!) - \frac{\epsilon N}{100T} + S$ bits. Thus, $S > \frac{\epsilon N}{100T}$.

Intuition for the encoding

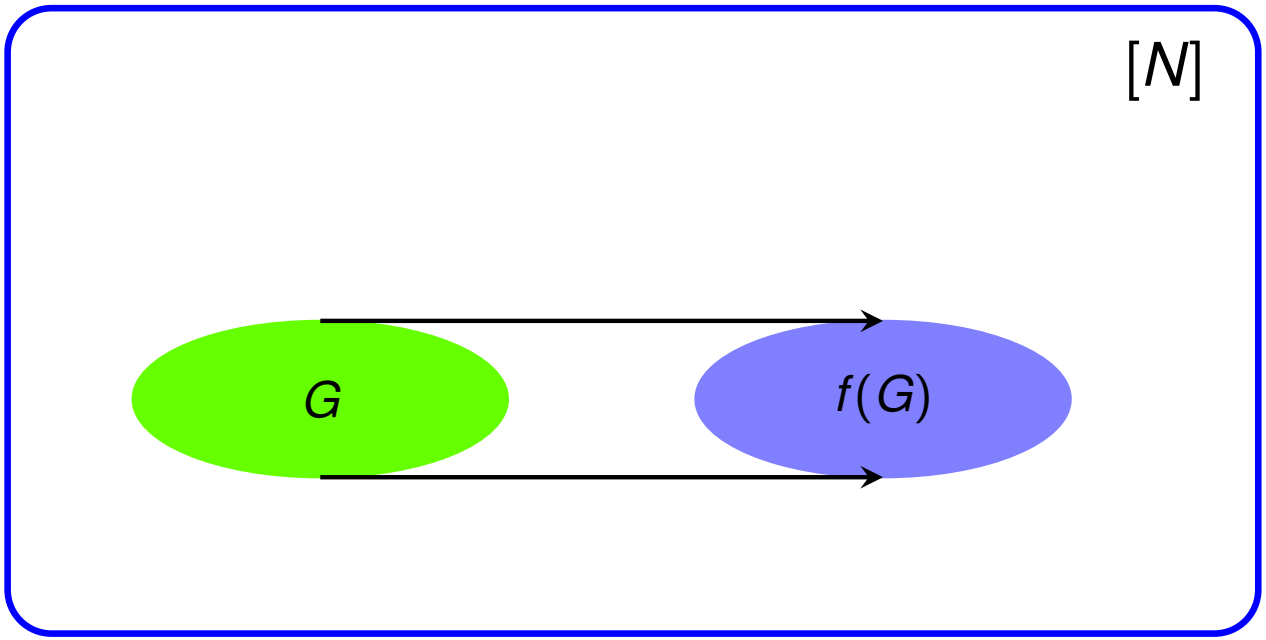
Intuition for the encoding



$$|G| = \frac{\epsilon N}{100T}$$

- A inverts G correctly.
- For all $x \in G$, A does not query any element in G.

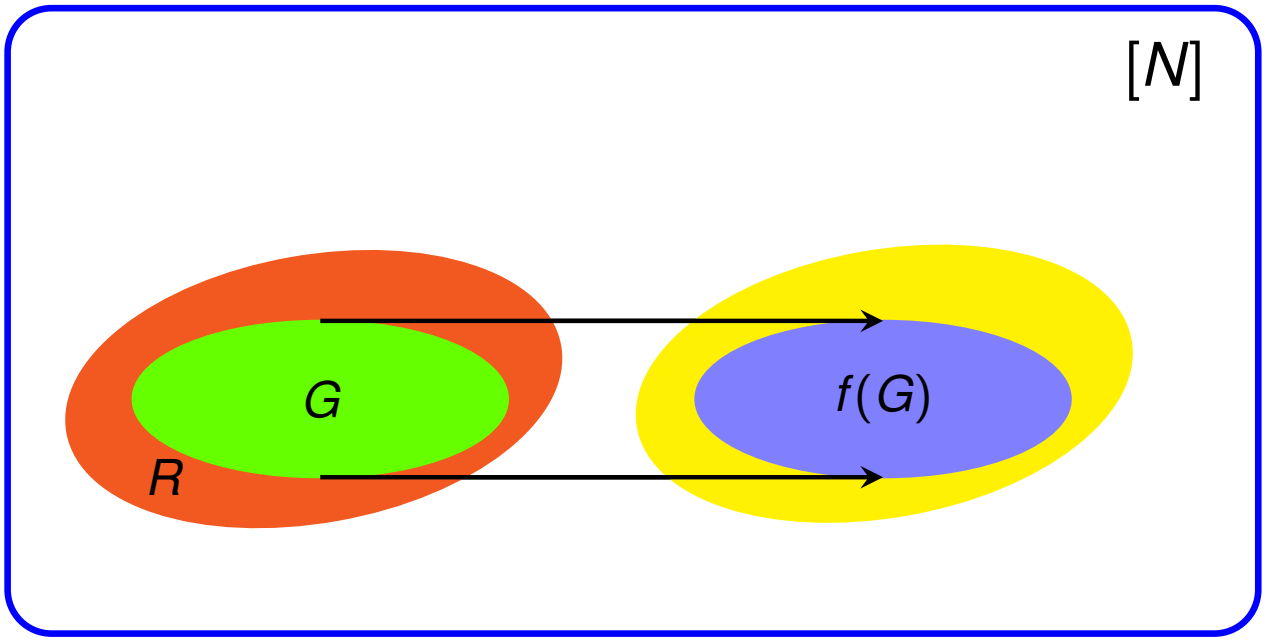
Intuition for the encoding



$$|G| = \frac{\epsilon N}{100T}$$

- A inverts G correctly.
- For all $x \in G$, A does not query any element in G.

Intuition for the encoding



$$|G| = \frac{\epsilon N}{100T}$$

$$|R| = \frac{N}{10T}$$

- A inverts G correctly.
- Set R is chosen at random. $G \subseteq R$.
- For all $x \in G$, A does not query any element in R.

Conclusions

- Non-uniform attacks can do better than uniform attacks on one-way functions and PRGs

Conclusions

- Non-uniform attacks can do better than uniform attacks on one-way functions and PRGs
- The best provable upper bound for one-way functions on all inputs remains $N^{3/4}$ and $N^{2/3}$ is the best for "Hellman"-style arguments (Barkan, Biham and Shamir)

Conclusions

- Non-uniform attacks can do better than uniform attacks on one-way functions and PRGs
- The best provable upper bound for one-way functions on all inputs remains $N^{3/4}$ and $N^{2/3}$ is the best for "Hellman"-style arguments (Barkan, Biham and Shamir)
- Techniques for proving lower bounds do not seem to do any better for one-way functions than permutations i.e. $\Omega(N^{1/2})$.

Thank You

Questions?