New and Improved Key-Homomorphic Pseudorandom Functions

Abhishek Banerjee[1] Chris Peikert[1]

[1] Georgia Institute of Technology

CRYPTO '14

19 August 2014

Outline

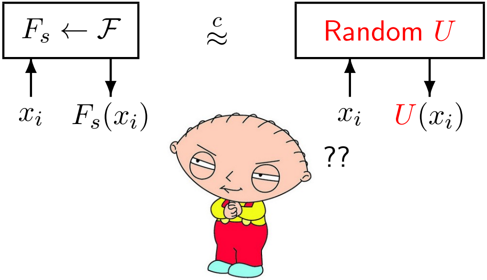
1. Introduction

2. Construction, Parameters and Efficiency

3. Proof of Security (Idea)

Outline

- A family of functions $\mathcal{F} = \{F_s : \{0,1\}^k \to B\}$ such that, given adaptive query access,



- Lots of applications in symmetric key cryptography: encryption, message authentication, friend or foe identification, . . .

(Thanks to Seth MacFarlane for the adversary)

Cooking a (Provably Secure) PRF

1. Goldreich-Goldwasser-Micali [GGM'84]
    - Based on any (doubling) PRG: $F_s(x_1, \ldots, x_k) = G_{x_k}(\cdots(G_{x_1}(s))\cdots)$

Cooking a (Provably Secure) PRF

1. Goldreich-Goldwasser-Micali [GGM'84]
   - Based on any (doubling) PRG: $F_s(x_1, \ldots, x_k) = G_{x_k}(\cdots(G_{x_1}(s))\cdots)$
2. Number-theoretic direct constructions [NR'97, NRR'00]
   - Framework: exponentiate to a product of (secret) exponents
   - Security from number-theoretic assumptions (DDH, factoring, . . .)

Cooking a (Provably Secure) PRF

1. Goldreich-Goldwasser-Micali [GGM'84]
   - Based on any (doubling) PRG: $F_s(x_1, \ldots, x_k) = G_{x_k}(\cdots(G_{x_1}(s))\cdots)$
2. Number-theoretic direct constructions [NR'97, NRR'00]
   - Framework: exponentiate to a product of (secret) exponents
   - Security from number-theoretic assumptions (DDH, factoring, ...)
3. Lattice-based direct constructions [BPR'12]
   - Framework: round a product of (secret) matrices/ring elements
   - Security from lattice assumptions (LWE, worst-case lattice problems)

Key-Homomorphic Pseudorandom Functions

Key Homomorphism

- Can efficiently compute $F_{s+t}(x)$ from $F_s(x)$ and $F_t(x)$
- Applications:

Key-Homomorphic Pseudorandom Functions

Key Homomorphism

- Can efficiently compute $F_{s+t}(x)$ from $F_s(x)$ and $F_t(x)$
- Applications: distribute the operation of a Key Distribution Center,

1. DDH-based construction [NPR'99]
   - Security in the random oracle model

Key-Homomorphic Pseudorandom Functions

Key Homomorphism

- Can efficiently compute $F_{s+t}(x)$ from $F_s(x)$ and $F_t(x)$
- Applications: distribute the operation of a Key Distribution Center, symmetric-key proxy re-encryption, updatable encryption, and PRFs secure against related-key attacks [BC'10,LMR'14]

1. DDH-based construction [NPR'99]
   - Security in the random oracle model
2. Lattice-based construction [BLMR'13]
   - Security in the standard model; construction and proof similar to [BPR'12] rounded-subset-product construction

Key-Homomorphic Pseudorandom Functions

Key Homomorphism

- Can efficiently compute $F_{s+t}(x)$ from $F_s(x)$ and $F_t(x)$
- Applications: distribute the operation of a Key Distribution Center, symmetric-key proxy re-encryption, updatable encryption, and PRFs secure against related-key attacks [BC'10,LMR'14]

1. DDH-based construction [NPR'99]
   - Security in the random oracle model
2. Lattice-based construction [BLMR'13]
   - Security in the standard model; construction and proof similar to [BPR'12] rounded-subset-product construction
   - Main drawback: has huge parameters, keys, and runtimes

Key-Homomorphic Pseudorandom Functions

Key Homomorphism

- Can efficiently compute $F_{s+t}(x)$ from $F_s(x)$ and $F_t(x)$
- Applications: distribute the operation of a Key Distribution Center, symmetric-key proxy re-encryption, updatable encryption, and PRFs secure against related-key attacks [BC'10,LMR'14]

1. DDH-based construction [NPR'99]
   - Security in the random oracle model
2. Lattice-based construction [BLMR'13]
   - Security in the standard model; construction and proof similar to [BPR'12] rounded-subset-product construction
   - Main drawback: has huge parameters, keys, and runtimes

- [BPR'12] also gives (non-KH) PRFs having much better parameters, with slightly worse (still polylog) depth

Key-Homomorphic Pseudorandom Functions

Key Homomorphism

- Can efficiently compute $F_{s+t}(x)$ from $F_s(x)$ and $F_t(x)$
- Applications: distribute the operation of a Key Distribution Center, symmetric-key proxy re-encryption, updatable encryption, and PRFs secure against related-key attacks [BC'10,LMR'14]

1. DDH-based construction [NPR'99]
   - Security in the random oracle model
2. Lattice-based construction [BLMR'13]
   - Security in the standard model; construction and proof similar to [BPR'12] rounded-subset-product construction
   - Main drawback: has huge parameters, keys, and runtimes

- [BPR'12] also gives (non-KH) PRFs having much better parameters, with slightly worse (still polylog) depth
- Can we obtain similar tradeoffs for KH-PRFs?

Our Results

⋆ New KH-PRFs (from lattices):
  - Polylog $\tilde{O}(1)$ depth (still)
  - Quasi-optimal $\tilde{O}(\lambda)$ key sizes

First sublinear-depth PRFs (KH or otherwise) with $\tilde{O}(\lambda)$ key size!

Our Results

- ⋆ New KH-PRFs (from lattices):
  - Polylog $\tilde{O}(1)$ depth (still)
  - Quasi-optimal $\tilde{O}(\lambda)$ key sizes

First sublinear-depth PRFs (KH or otherwise) with $\tilde{O}(\lambda)$ key size!

Reference	**Key**	**Pub Params**	**Time/Bit**
[BLMR'13]	λ^3 $[\lambda^3]$	λ^6 $[\lambda^4]$	λ^5 $[\lambda^3]$
This work	λ $[\lambda]$	λ^2 $[\lambda]$	λ^ω $[\lambda]$

Figure : For input length λ with 2^λ security under standard assumptions. Log factors omitted. Ring-based constructions appear in [brackets].

Our Results

- ⋆ New KH-PRFs (from lattices):
  - Polylog $\tilde{O}(1)$ depth (still)
  - Quasi-optimal $\tilde{O}(\lambda)$ key sizes

First sublinear-depth PRFs (KH or otherwise) with $\tilde{O}(\lambda)$ key size!

Reference	**Key**	**Pub Params**	**Time/Bit**
[BLMR'13]	λ^3 $[\lambda^3]$	λ^6 $[\lambda^4]$	λ^5 $[\lambda^3]$
This work	λ $[\lambda]$	λ^2 $[\lambda]$	λ^ω $[\lambda]$

Figure : For input length λ with 2^λ security under standard assumptions. Log factors omitted. Ring-based constructions appear in [brackets].

- ⋆ New proof technique that may be useful elsewhere

Our Results

- ⋆ New KH-PRFs (from lattices):
  - Polylog $\tilde{O}(1)$ depth (still)
  - Quasi-optimal $\tilde{O}(\lambda)$ key sizes

First sublinear-depth PRFs (KH or otherwise) with $\tilde{O}(\lambda)$ key size!

Reference	**Key**	**Pub Params**	**Time/Bit**
[BLMR'13]	λ^3 $[\lambda^3]$	λ^6 $[\lambda^4]$	λ^5 $[\lambda^3]$
This work	λ $[\lambda]$	λ^2 $[\lambda]$	λ^ω $[\lambda]$

Figure : For input length λ with 2^λ security under standard assumptions. Log factors omitted. Ring-based constructions appear in [brackets].

- ⋆ New proof technique that may be useful elsewhere

Full version: `http://eprint.iacr.org/2014/074`

Outline

1. Introduction
2. Construction, Parameters and Efficiency
3. Proof of Security (Idea)

Boneh *et al.* KH-PRF Construction [BLMR'13]

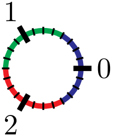
- Secret key $\mathbf{s} \in \mathbb{Z}_q^n$, pub params $\mathbf{B}_0, \mathbf{B}_1 \in \{0,1\}^{n \times n}$, input $x \in \{0,1\}^k$

$$F_{\mathbf{s}}(x) = \left\lfloor \mathbf{s} \cdot \prod_{i=1}^{k} \mathbf{B}_{x_i} \right\rceil_p$$

Boneh *et al.* KH-PRF Construction [BLMR'13]

- Secret key $\mathbf{s} \in \mathbb{Z}_q^n$, pub params $\mathbf{B}_0, \mathbf{B}_1 \in \{0,1\}^{n \times n}$, input $x \in \{0,1\}^k$

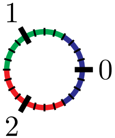
$$F_{\mathbf{s}}(x) = \left\lfloor \mathbf{s} \cdot \prod_{i=1}^{k} \mathbf{B}_{x_i} \right\rceil_p$$



Boneh *et al.* KH-PRF Construction [BLMR'13]

- Secret key $\mathbf{s} \in \mathbb{Z}_q^n$, pub params $\mathbf{B}_0, \mathbf{B}_1 \in \{0,1\}^{n \times n}$, input $x \in \{0,1\}^k$

$$F_{\mathbf{s}}(x) = \left\lfloor \mathbf{s} \cdot \prod_{i=1}^{k} \mathbf{B}_{x_i} \right\rceil_p$$

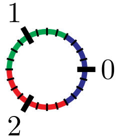


- "Somewhat key-homomorphic:" $F_{\mathbf{s}}(x) + F_{\mathbf{t}}(x) \in F_{\mathbf{s}+\mathbf{t}}(x) + \{0, \pm 1\}^n$

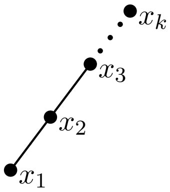
Boneh *et al.* KH-PRF Construction [BLMR'13]

- Secret key $\mathbf{s} \in \mathbb{Z}_q^n$, pub params $\mathbf{B}_0, \mathbf{B}_1 \in \{0,1\}^{n \times n}$, input $x \in \{0,1\}^k$

$$F_{\mathbf{s}}(x) = \left\lfloor \mathbf{s} \cdot \prod_{i=1}^{k} \mathbf{B}_{x_i} \right\rceil_p$$



- "Somewhat key-homomorphic:" $F_{\mathbf{s}}(x) + F_{\mathbf{t}}(x) \in F_{\mathbf{s}+\mathbf{t}}(x) + \{0, \pm 1\}^n$
- Proof strategy: introduce "short" error which "rounds away"

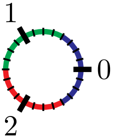


$$F_{\mathbf{s}}(x) = \left\lfloor \mathbf{s} \cdot \prod_{i=1}^{k} \mathbf{B}_{x_i} \right\rceil_p \stackrel{s}{\approx} \left\lfloor \underbrace{(\mathbf{s}\mathbf{B}_{x_1} + \mathbf{e}_{x_1})}_{\mathbf{s}_{x_1}} \cdot \prod_{i=2}^{k} \mathbf{B}_{x_i} \right\rceil_p$$

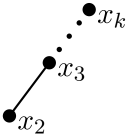
Boneh *et al.* KH-PRF Construction [BLMR'13]

- Secret key $\mathbf{s} \in \mathbb{Z}_q^n$, pub params $\mathbf{B}_0, \mathbf{B}_1 \in \{0,1\}^{n \times n}$, input $x \in \{0,1\}^k$

$$F_{\mathbf{s}}(x) = \left\lfloor \mathbf{s} \cdot \prod_{i=1}^{k} \mathbf{B}_{x_i} \right\rceil_p$$



- "Somewhat key-homomorphic:" $F_{\mathbf{s}}(x) + F_{\mathbf{t}}(x) \in F_{\mathbf{s}+\mathbf{t}}(x) + \{0, \pm 1\}^n$
- Proof strategy: introduce "short" error which "rounds away"



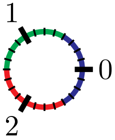
$$F_{\mathbf{s}}(x) = \left\lfloor \mathbf{s} \cdot \prod_{i=1}^{k} \mathbf{B}_{x_i} \right\rceil_p \stackrel{s}{\approx} \left\lfloor \underbrace{(\mathbf{s}\mathbf{B}_{x_1} + \mathbf{e}_{x_1})}_{\mathbf{s}_{x_1}} \cdot \prod_{i=2}^{k} \mathbf{B}_{x_i} \right\rceil_p$$

$$\stackrel{c}{\approx} \left\lfloor \mathbf{s}_{x_1} \cdot \prod_{i=2}^{k} \mathbf{B}_{x_i} \right\rceil_p \stackrel{c}{\approx} \ldots \stackrel{c}{\approx} \lfloor \mathbf{s}_x \rceil_p = U(x)$$

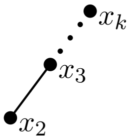
Boneh *et al.* KH-PRF Construction [BLMR'13]

- Secret key $\mathbf{s} \in \mathbb{Z}_q^n$, pub params $\mathbf{B}_0, \mathbf{B}_1 \in \{0,1\}^{n \times n}$, input $x \in \{0,1\}^k$

$$F_{\mathbf{s}}(x) = \left\lfloor \mathbf{s} \cdot \prod_{i=1}^{k} \mathbf{B}_{x_i} \right\rceil_p$$



- "Somewhat key-homomorphic:" $F_{\mathbf{s}}(x) + F_{\mathbf{t}}(x) \in F_{\mathbf{s}+\mathbf{t}}(x) + \{0, \pm 1\}^n$
- Proof strategy: introduce "short" error which "rounds away"



$$F_{\mathbf{s}}(x) = \left\lfloor \mathbf{s} \cdot \prod_{i=1}^{k} \mathbf{B}_{x_i} \right\rceil_p \overset{s}{\approx} \left\lfloor \underbrace{(\mathbf{s}\mathbf{B}_{x_1} + \mathbf{e}_{x_1})}_{\mathbf{s}_{x_1}} \cdot \prod_{i=2}^{k} \mathbf{B}_{x_i} \right\rceil_p$$

$$\overset{c}{\approx} \left\lfloor \mathbf{s}_{x_1} \cdot \prod_{i=2}^{k} \mathbf{B}_{x_i} \right\rceil_p \overset{c}{\approx} \dots \overset{c}{\approx} \lfloor \mathbf{s}_x \rceil_p = U(x)$$

✗ LWE approx factor grows exponentially in input length k.

Gadget and Bit-Decomposition

- "Gadget" $\mathbb{Z}_q$-matrix $\mathbf{G}$ [MP'12]:

$$\underbrace{\begin{bmatrix} \quad \mathbf{A} \quad \end{bmatrix}}_{\substack{\textit{Any} \\ \mathbb{Z}_q\text{-matrix}}} = \begin{bmatrix} \quad \mathbf{G} \quad \end{bmatrix} \cdot \underbrace{\begin{bmatrix} \\ \mathbf{G}^{-1}(\mathbf{A}) \\ \\ \end{bmatrix}}_{\substack{\textit{Square} \\ \{0,1\}\text{-matrix}}}$$

Gadget and Bit-Decomposition

- "Gadget" $\mathbb{Z}_q$-matrix $\mathbf{G}$ [MP'12]:

$$\underbrace{\begin{bmatrix} \quad \mathbf{A} \quad \end{bmatrix}}_{\textit{Any } \mathbb{Z}_q\text{-matrix}} = \begin{bmatrix} \quad \mathbf{G} \quad \end{bmatrix} \cdot \underbrace{\begin{bmatrix} \mathbf{G}^{-1}(\mathbf{A}) \end{bmatrix}}_{\textit{Square } \{0,1\}\text{-matrix}}$$

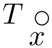
- A ubiquitous tool in lattice cryptography: FHE [BV'11,GSW'13,AP'14], CCA/IBE/ABE/FHS [MP'12,BGG$^+$'14,GVW'14]

Our Construction

For matrices $\mathbf{A}_0, \mathbf{A}_1$, full binary tree T and $x \in \{0,1\}^{|T|}$, define $\mathbf{A}_T(x)$:

Our Construction

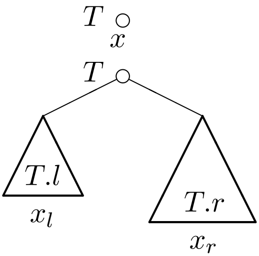
For matrices $\mathbf{A}_0, \mathbf{A}_1$, full binary tree T and $x \in \{0,1\}^{|T|}$, define $\mathbf{A}_T(x)$:



$$\mathbf{A}_T(x) := \mathbf{A}_x \quad \text{for } |T| = 1$$

Our Construction

For matrices $\mathbf{A}_0, \mathbf{A}_1$, full binary tree T and $x \in \{0,1\}^{|T|}$, define $\mathbf{A}_T(x)$:

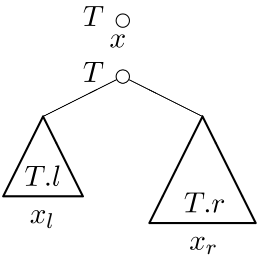


$$\mathbf{A}_T(x) := \mathbf{A}_x \quad \text{for } |T| = 1$$

$$\mathbf{A}_T(x_l \| x_r) := \mathbf{A}_{T.l}(x_l) \cdot \mathbf{G}^{-1}(\mathbf{A}_{T.r}(x_r))$$

Our Construction

For matrices $\mathbf{A}_0, \mathbf{A}_1$, full binary tree T and $x \in \{0,1\}^{|T|}$, define $\mathbf{A}_T(x)$:



$$\mathbf{A}_T(x) := \mathbf{A}_x \quad \text{for } |T| = 1$$

$$\mathbf{A}_T(x_l \| x_r) := \mathbf{A}_{T.l}(x_l) \cdot \mathbf{G}^{-1}(\mathbf{A}_{T.r}(x_r))$$

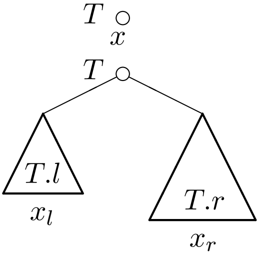
New KH-PRF Construction

- Public parameters: matrices $\mathbf{A}_0, \mathbf{A}_1$, full binary tree T
- Function $F_{\mathbf{s}}$ on $|T|$-bit input x defined as

$$F_{\mathbf{s}}(x) = \lfloor \mathbf{s} \cdot \mathbf{A}_T(x) \rceil_p$$

Our Construction

For matrices $\mathbf{A}_0, \mathbf{A}_1$, full binary tree T and $x \in \{0,1\}^{|T|}$, define $\mathbf{A}_T(x)$:



$$\mathbf{A}_T(x) := \mathbf{A}_x \quad \text{for } |T| = 1$$

$$\mathbf{A}_T(x_l \| x_r) := \mathbf{A}_{T.l}(x_l) \cdot \mathbf{G}^{-1}(\mathbf{A}_{T.r}(x_r))$$

New KH-PRF Construction

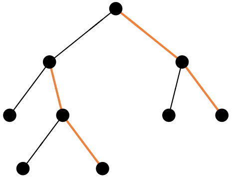
- Public parameters: matrices $\mathbf{A}_0, \mathbf{A}_1$, full binary tree T
- Function $F_{\mathbf{s}}$ on $|T|$-bit input x defined as

$$F_{\mathbf{s}}(x) = \lfloor \mathbf{s} \cdot \mathbf{A}_T(x) \rceil_p$$

- Somewhat KH just as in [BLMR'13]. Same applications!

Parameters and Parallelism

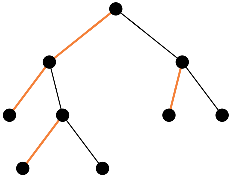
- Sequentiality $s(T)$: the "right depth" of T
  - Circuit depth of PRF is proportional to $s(T)$



$s = 2$

Parameters and Parallelism

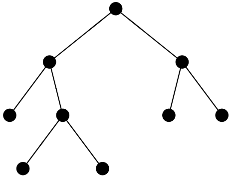
- Sequentiality $s(T)$: the "right depth" of T
  - Circuit depth of PRF is proportional to $s(T)$
- Expansion $e(T)$: the "left depth" of T
  - LWE approx factor is exponential in $e(T)$



$e = 2$

Parameters and Parallelism

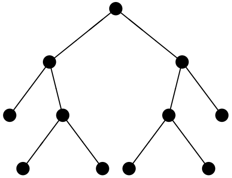
- Sequentiality $s(T)$: the "right depth" of T
  - Circuit depth of PRF is proportional to $s(T)$
- Expansion $e(T)$: the "left depth" of T
  - LWE approx factor is exponential in $e(T)$
- Max input length = max # leaves = $\binom{e+s}{s}$



$s = 2$, $e = 2$

Parameters and Parallelism

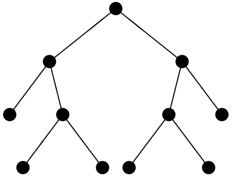
- Sequentiality $s(T)$: the "right depth" of T
  - Circuit depth of PRF is proportional to $s(T)$
- Expansion $e(T)$: the "left depth" of T
  - LWE approx factor is exponential in $e(T)$
- Max input length = max # leaves = $\binom{e+s}{s}$



$s = 2$, $e = 2$

Parameters and Parallelism

- Sequentiality $s(T)$: the "right depth" of T
  - Circuit depth of PRF is proportional to $s(T)$
- Expansion $e(T)$: the "left depth" of T
  - LWE approx factor is exponential in $e(T)$
- Max input length = max # leaves = $\binom{e+s}{s}$

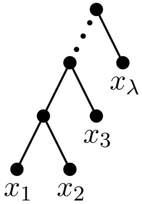


$s = 2$, $e = 2$

Instantiations

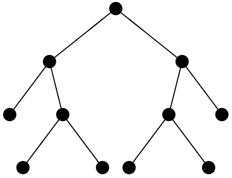
$e(T)$	$s(T)$	Key	Params
$\lambda - 1$	1	λ^3	λ^6

"Left Spine"



Parameters and Parallelism

- Sequentiality $s(T)$: the "right depth" of T
  - Circuit depth of PRF is proportional to $s(T)$
- Expansion $e(T)$: the "left depth" of T
  - LWE approx factor is exponential in $e(T)$
- Max input length = max # leaves = $\binom{e+s}{s}$



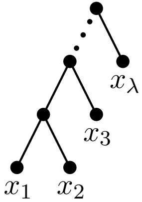
$s = 2,\ e = 2$

Instantiations

$e(T)$	$s(T)$	Key	Params
$\lambda - 1$	1	λ^3	λ^6

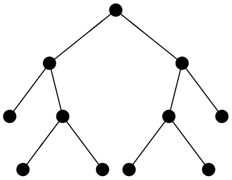
[BLMR'13]

Construction!



Parameters and Parallelism

- Sequentiality $s(T)$: the "right depth" of T
  - Circuit depth of PRF is proportional to $s(T)$
- Expansion $e(T)$: the "left depth" of T
  - LWE approx factor is exponential in $e(T)$
- Max input length $=$ max # leaves $= \binom{e+s}{s}$

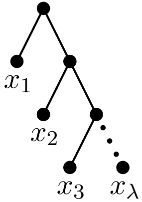


$s = 2$, $e = 2$

Instantiations

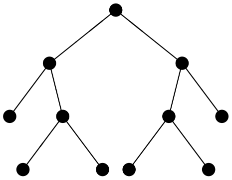
$e(T)$	$s(T)$	Key	Params
$\lambda - 1$	1	λ^3	λ^6
1	$\lambda - 1$	λ	λ^2

"Right Spine"



Parameters and Parallelism

- Sequentiality $s(T)$: the "right depth" of T
  - Circuit depth of PRF is proportional to $s(T)$
- Expansion $e(T)$: the "left depth" of T
  - LWE approx factor is exponential in $e(T)$
- Max input length = max # leaves = $\binom{e+s}{s}$



$s = 2,\ e = 2$

Instantiations

$e(T)$	$s(T)$	Key	Params
$\lambda - 1$	1	λ^3	λ^6
1	$\lambda - 1$	λ	λ^2
$\approx \log_4(\lambda)$	$\approx \log_4(\lambda)$	λ	λ^2

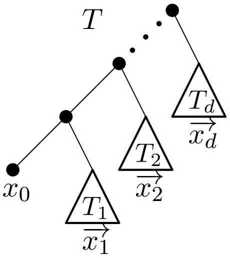
Outline

1. Introduction

2. Construction, Parameters and Efficiency

3. Proof of Security (Idea)

Proof Idea



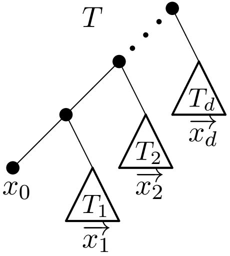
$$F_{\mathbf{s}}(x) = \left\lfloor \mathbf{s} \cdot \mathbf{A}_{x_0} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

$$\stackrel{s}{\approx} \left\lfloor \underbrace{(\mathbf{s} \cdot \mathbf{A}_{x_0} + \mathbf{e}_{x_0})}_{\mathbf{u}_{x_0}} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

$$\stackrel{c}{\approx} \left\lfloor \mathbf{u}_{x_0} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

Proof Idea

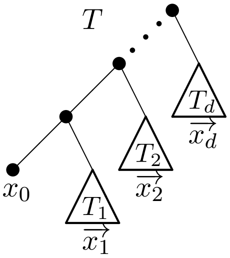
✓ New Idea: $\mathbf{u} = \mathbf{s} \cdot \mathbf{G} + \mathbf{v}$ for uniform, *independent* $\mathbf{s}$ and $\mathbf{v} \in \mathcal{P}(\mathbf{G})$.



$$
\begin{aligned}
F_{\mathbf{s}}(x) &= \left\lfloor \mathbf{s} \cdot \mathbf{A}_{x_0} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p \\
&\overset{s}{\approx} \left\lfloor \underbrace{(\mathbf{s} \cdot \mathbf{A}_{x_0} + \mathbf{e}_{x_0})}_{\mathbf{u}_{x_0}} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p \\
&\overset{c}{\approx} \left\lfloor \mathbf{u}_{x_0} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p
\end{aligned}
$$

Proof Idea

✓ New Idea: $\mathbf{u} = \mathbf{s} \cdot \mathbf{G} + \mathbf{v}$ for uniform, *independent* $\mathbf{s}$ and $\mathbf{v} \in \mathcal{P}(\mathbf{G})$.



$$F_{\mathbf{s}}(x) = \left\lfloor \mathbf{s} \cdot \mathbf{A}_{x_0} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

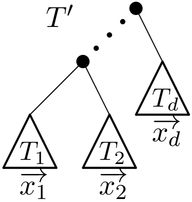
$$\overset{s}{\approx} \left\lfloor \underbrace{(\mathbf{s} \cdot \mathbf{A}_{x_0} + \mathbf{e}_{x_0})}_{\mathbf{u}_{x_0}} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

$$\overset{c}{\approx} \left\lfloor \mathbf{u}_{x_0} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

$$= \left\lfloor \mathbf{s}_{x_0} \cdot \mathbf{A}_{T_1}(\overrightarrow{x_1}) \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_2}(\overrightarrow{x_2})) \cdots + \mathbf{v}_{x_0} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

Proof Idea

✓ New Idea: $\mathbf{u} = \mathbf{s} \cdot \mathbf{G} + \mathbf{v}$ for uniform, *independent* $\mathbf{s}$ and $\mathbf{v} \in \mathcal{P}(\mathbf{G})$.



$$F_{\mathbf{s}}(x) = \left\lfloor \mathbf{s} \cdot \mathbf{A}_{x_0} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

$$\overset{s}{\approx} \left\lfloor \underbrace{(\mathbf{s} \cdot \mathbf{A}_{x_0} + \mathbf{e}_{x_0})}_{\mathbf{u}_{x_0}} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

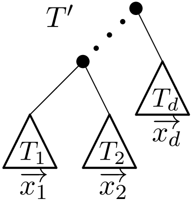
$$\overset{c}{\approx} \left\lfloor \mathbf{u}_{x_0} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

$$= \left\lfloor \mathbf{s}_{x_0} \cdot \mathbf{A}_{T_1}(\overrightarrow{x_1}) \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_2}(\overrightarrow{x_2})) \cdots + \mathbf{v}_{x_0} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

$$= \left\lfloor \mathbf{s}_{x_0} \cdot \mathbf{A}_{T'}(\overrightarrow{x_1} \| \cdots \| \overrightarrow{x_d}) + \mathbf{v}_{x_0} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

Proof Idea

✓ New Idea: $\mathbf{u} = \mathbf{s} \cdot \mathbf{G} + \mathbf{v}$ for uniform, *independent* $\mathbf{s}$ and $\mathbf{v} \in \mathcal{P}(\mathbf{G})$.



$$F_{\mathbf{s}}(x) = \left\lfloor \mathbf{s} \cdot \mathbf{A}_{x_0} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

$$\stackrel{s}{\approx} \left\lfloor \underbrace{(\mathbf{s} \cdot \mathbf{A}_{x_0} + \mathbf{e}_{x_0})}_{\mathbf{u}_{x_0}} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

$$\stackrel{c}{\approx} \left\lfloor \mathbf{u}_{x_0} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

$$= \left\lfloor \mathbf{s}_{x_0} \cdot \mathbf{A}_{T_1}(\overrightarrow{x_1}) \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_2}(\overrightarrow{x_2})) \cdots + \mathbf{v}_{x_0} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

$$= \left\lfloor \mathbf{s}_{x_0} \cdot \mathbf{A}_{T'}(\overrightarrow{x_1} \| \cdots \| \overrightarrow{x_d}) + \mathbf{v}_{x_0} \cdot \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots \right\rceil_p$$

$$\cdots \stackrel{c}{\approx} \left\lfloor \mathbf{s}_x + \mathbf{v}_{x_0} \mathbf{G}^{-1}(\mathbf{A}_{T_1}(\overrightarrow{x_1})) \cdots + \text{other } \mathbf{v} \text{ terms} \right\rceil_p \stackrel{s}{\approx} U(x). \qquad \blacksquare$$

Conclusions

Our main contributions

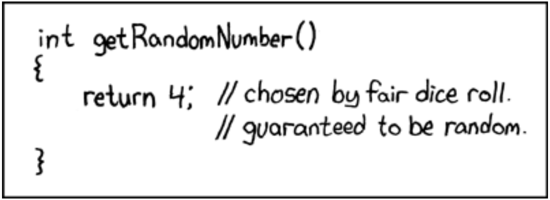
- New KH-PRFs from lattices: quasi-optimal key sizes, polylog depth
- New proof technique

Conclusions

Our main contributions

- New KH-PRFs from lattices: quasi-optimal key sizes, polylog depth
- New proof technique

The Last Word [Mun'07]



(Image source: http://xkcd.com/221/)