Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices

Chris Peikert[1] Alon Rosen[2]

[1]MIT CSAIL

[2]Harvard DEAS

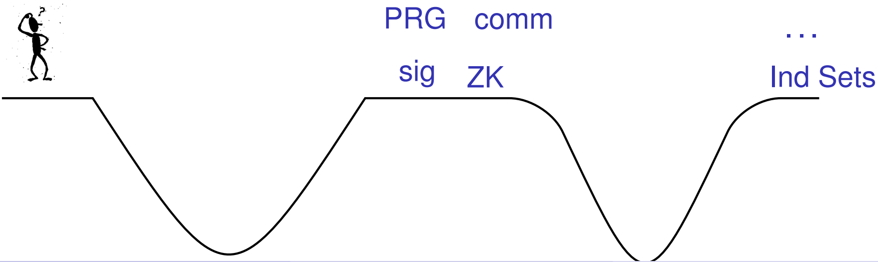
Theory of Cryptography Conference

5 March 2006

One-Way Function (family):

$$a, \quad y = f_a(x) \xrightarrow{\text{hard}} x' \in f_a^{-1}(y)$$

✔ Sufficient for *some* crypto

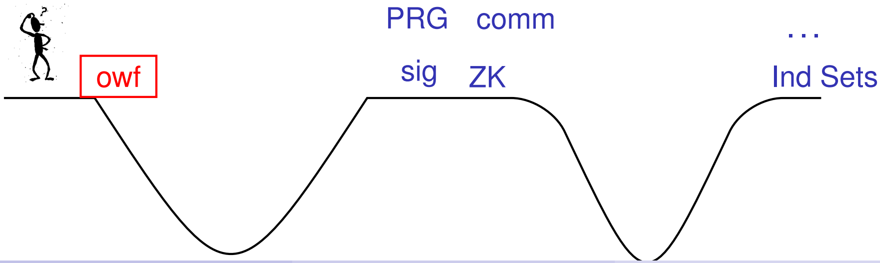


One-Wayness vs. Collision-Resistance

One-Way Function (family):

$$a, \quad y = f_a(x) \xrightarrow{\text{hard}} x' \in f_a^{-1}(y)$$

- ✔ Sufficient for *some* crypto
- ✗ But applications use OWFs *inefficiently*...
  This is inherent (black-box)! [GeTr, GGK, HoKa]

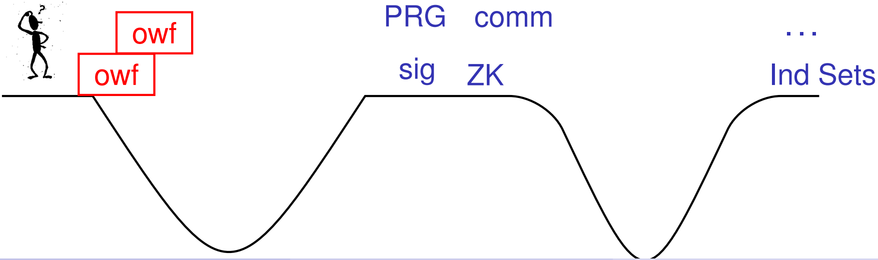


One-Way Function (family):

$$a, \quad y = f_a(x) \xrightarrow{\text{hard}} x' \in f_a^{-1}(y)$$

✔ Sufficient for *some* crypto

✗ But applications use OWFs *inefficiently*...
This is inherent (black-box)! [GeTr, GGK, HoKa]



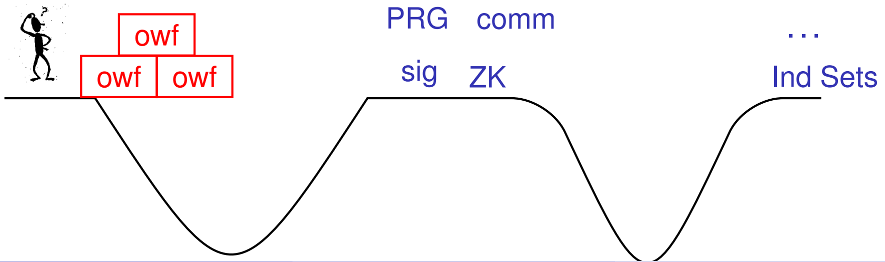
One-Wayness vs. Collision-Resistance

One-Way Function (family):

$$a, \quad y = f_a(x) \xrightarrow{\text{hard}} x' \in f_a^{-1}(y)$$

✔ Sufficient for *some* crypto

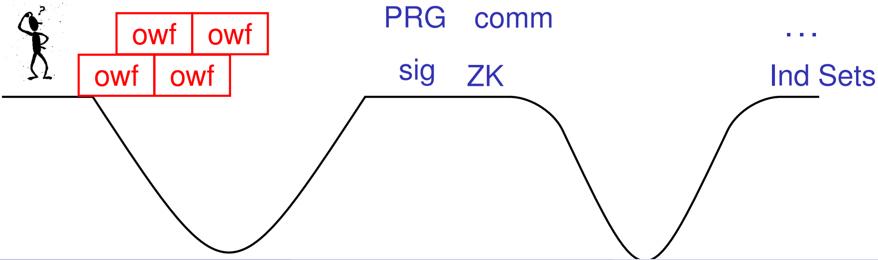
✗ But applications use OWFs *inefficiently*...
This is inherent (black-box)! [GeTr, GGK, HoKa]



One-Way Function (family):

$$a, \quad y = f_a(x) \xrightarrow{\text{hard}} x' \in f_a^{-1}(y)$$

- ✔ Sufficient for *some* crypto
- ✗ But applications use OWFs *inefficiently*...
  This is inherent (black-box)! [GeTr, GGK, HoKa]

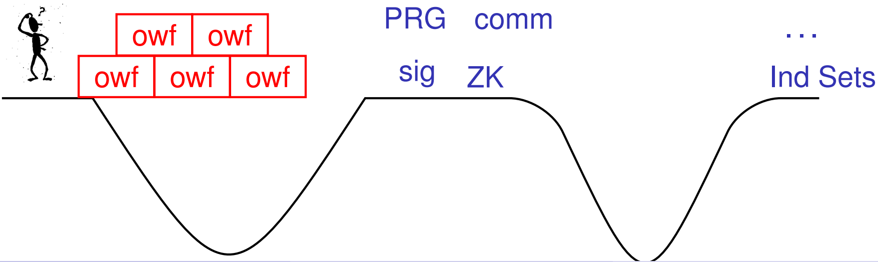


One-Wayness vs. Collision-Resistance

One-Way Function (family):

$$a, \quad y = f_a(x) \xrightarrow{\text{hard}} x' \in f_a^{-1}(y)$$

- ✔ Sufficient for *some* crypto
- ✗ But applications use OWFs *inefficiently*...
  This is inherent (black-box)! [GeTr, GGK, HoKa]



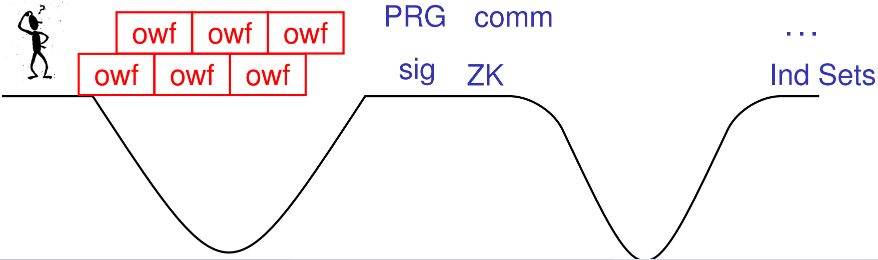
One-Wayness vs. Collision-Resistance

One-Way Function (family):

$$a, \quad y = f_a(x) \xrightarrow{\text{hard}} x' \in f_a^{-1}(y)$$

✔ Sufficient for *some* crypto

✗ But applications use OWFs *inefficiently*...
This is inherent (black-box)! [GeTr, GGK, HoKa]



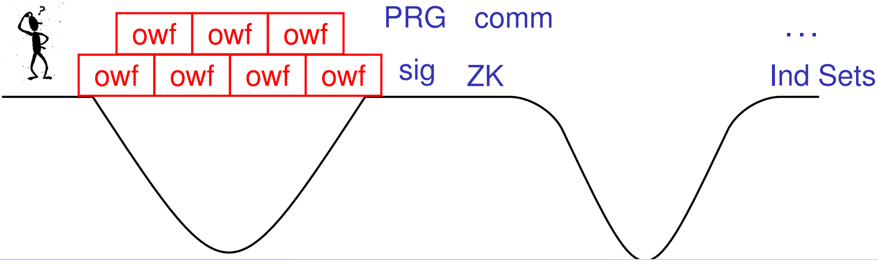
One-Wayness vs. Collision-Resistance

One-Way Function (family):

$$a, \quad y = f_a(x) \xrightarrow{\text{hard}} x' \in f_a^{-1}(y)$$

✔ Sufficient for *some* crypto

✗ But applications use OWFs *inefficiently*...
This is inherent (black-box)! [GeTr, GGK, HoKa]

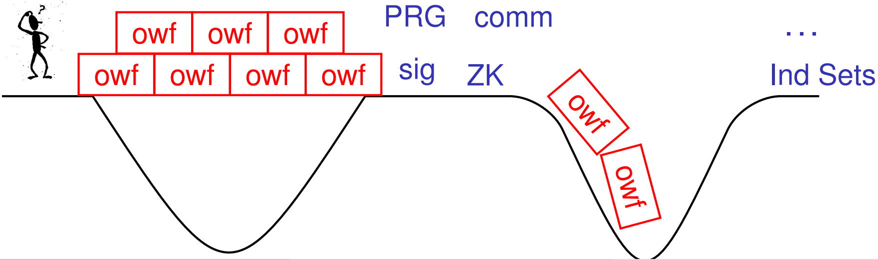


One-Wayness vs. Collision-Resistance

One-Way Function (family):

$$a, \quad y = f_a(x) \xrightarrow{\text{hard}} x' \in f_a^{-1}(y)$$

- ✔ Sufficient for *some* crypto
- ✗ But applications use OWFs *inefficiently*...
  This is inherent (black-box)! [GeTr, GGK, HoKa]
- ✗ Can't realize some notions at all! (black-box)

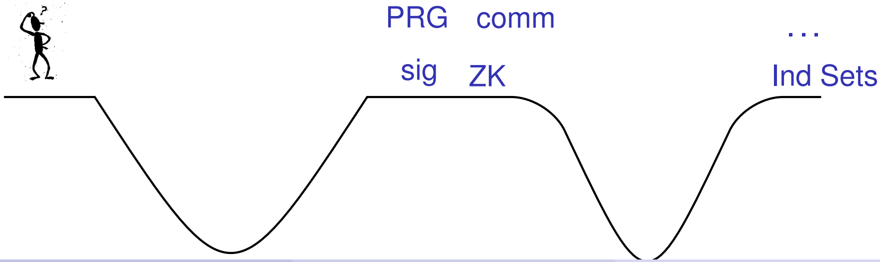


One-Wayness vs. Collision-Resistance

Collision-Resistant Hash (family):

$$a \xrightarrow{\text{hard}} x, x' \ : \ f_a(x) = f_a(x')$$

✔ Can construct more applications

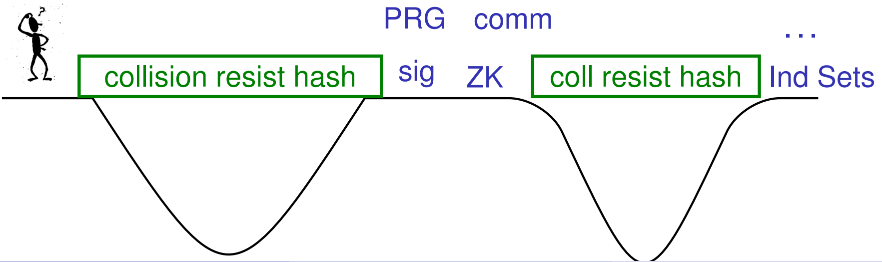


One-Wayness vs. Collision-Resistance

Collision-Resistant Hash (family):

$$a \xrightarrow{\text{hard}} x, x' \ : \ f_a(x) = f_a(x')$$

- ✔ Can construct more applications
- ✔ Applications use hashing efficiently!

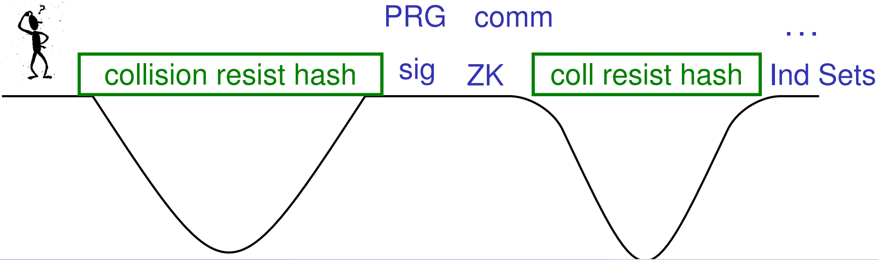


One-Wayness vs. Collision-Resistance

Collision-Resistant Hash (family):

$$a \xrightarrow{\text{hard}} x, x' \ : \ f_a(x) = f_a(x')$$

- ✔ Can construct more applications
- ✔ Applications use hashing efficiently!
- ?? BUT: is the hash itself efficient?

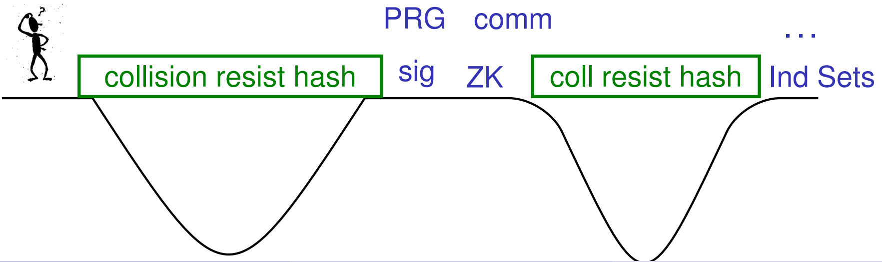


One-Wayness vs. Collision-Resistance

Collision-Resistant Hash (family):

$$a \xrightarrow{\text{hard}} x, x' \;:\; f_a(x) = f_a(x')$$

- ✔ Can construct more applications
- ✔ Applications use hashing efficiently!
- ?? BUT: is the hash *itself* efficient?
- ☞ MD5, SHA-1 highlight need for sound & efficient hashes



Our Contributions

Hash Function

- ✔ Very efficient: evaluate with just a few FFTs
- ✔ Collision-resistant: worst-case assumption on cyclic lattices
- ✔ Tighter & simpler security reduction than related works

Our Contributions

Hash Function

- ✔ Very efficient: evaluate with just a few FFTs
- ✔ Collision-resistant: worst-case assumption on cyclic lattices
- ✔ Tighter & simpler security reduction than related works

Understanding

- ✔ New algebraic interpretation of cyclic lattices
- ✔ New and tight connections among problems on cyclic lattices

Our Contributions

Hash Function

- ✔ Very efficient: evaluate with just a few FFTs
- ✔ Collision-resistant: worst-case assumption on cyclic lattices
- ✔ Tighter & simpler security reduction than related works

Understanding

- ✔ New algebraic interpretation of cyclic lattices
- ✔ New and tight connections among problems on cyclic lattices

☞ Our function is a certain kind of knapsack…

Generalized Knapsack Function [Mic02]

Let R be a ring with $+$ and $\times$, and let $S \subseteq R$. For:

- $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_m) \in R^m$ — m "weights": key
- $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_m) \in S^m$ — m "coeffs": input

$$f_{\mathbf{A}}(\mathbf{X}) = \sum_{i=1}^{m} \mathbf{a}_i \times \mathbf{x}_i$$

Generalized Knapsack Function [Mic02]

Let R be a ring with $+$ and $\times$, and let $S \subseteq R$. For:

- $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_m) \in R^m$ — m "weights": key
- $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_m) \in S^m$ — m "coeffs": input

$$f_{\mathbf{A}}(\mathbf{X}) = \sum_{i=1}^{m} \mathbf{a}_i \times \mathbf{x}_i$$

☞ Efficiency determined by m ("width"); runtime of $\times$, $+$.

Generalized Knapsack Function [Mic02]

Let R be a ring with $+$ and $\times$, and let $S \subseteq R$. For:

- $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_m) \in R^m$ — m "weights": key
- $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_m) \in S^m$ — m "coeffs": input

$$f_{\mathbf{A}}(\mathbf{X}) = \sum_{i=1}^{m} \mathbf{a}_i \times \mathbf{x}_i$$

☞ Efficiency determined by m ("width"); runtime of $\times$, $+$.

Lineage of Cryptographic Knapsacks

Knapsack Function	Security Notion	Efficient?
[Ajt96, GGH97]	collision-resistant	✗
[Mic02]	one-way	✔
Today	collision-resistant	✔✔

Micciancio's Function

- $R = (\mathbb{Z}_p^n, +, \otimes)$, where $\otimes$ is cyclic convolution:

$$\begin{bmatrix} | \\ \mathbf{a} \\ | \end{bmatrix} \otimes \begin{bmatrix} | \\ \mathbf{x} \\ | \end{bmatrix} = \begin{bmatrix} a_0 & a_{n-1} & \cdots & a_1 \\ a_1 & a_0 & \cdots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

Micciancio's Function

- $R = (\mathbb{Z}_p^n, +, \otimes)$, where $\otimes$ is cyclic convolution:

$$\begin{bmatrix} | \\ \mathbf{a} \\ | \end{bmatrix} \otimes \begin{bmatrix} | \\ \mathbf{x} \\ | \end{bmatrix} = \begin{bmatrix} a_0 & a_{n-1} & \cdots & a_1 \\ a_1 & a_0 & \cdots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

- $S = \{\mathbf{x} \in R \ : \ \|\mathbf{x}\|_\infty$ is small$\}$. (Note: $|S|$ is exponential in n.)

Micciancio's Function

- $R = (\mathbb{Z}_p^n, +, \otimes)$, where $\otimes$ is cyclic convolution:

$$\begin{bmatrix} | \\ \mathbf{a} \\ | \end{bmatrix} \otimes \begin{bmatrix} | \\ \mathbf{x} \\ | \end{bmatrix} = \begin{bmatrix} a_0 & a_{n-1} & \cdots & a_1 \\ a_1 & a_0 & \cdots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

- $S = \{\mathbf{x} \in R \ : \ \|\mathbf{x}\|_\infty$ is small$\}$. (Note: $|S|$ is exponential in n.)

Evaluating f

$$\mathbf{A} = \begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_m \\ | & | & & | \end{bmatrix} \in R^m$$

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_m \\ | & | & & | \end{bmatrix} \in S^m$$

$$f_{\mathbf{A}}(\mathbf{X}) = \sum_i \begin{bmatrix} | \\ \mathbf{a}_i \\ | \end{bmatrix} \otimes \begin{bmatrix} | \\ \mathbf{x}_i \\ | \end{bmatrix}$$

Micciancio's Function

- $R = (\mathbb{Z}_p^n, +, \otimes)$, where $\otimes$ is cyclic convolution:

$$\begin{bmatrix} | \\ \mathbf{a} \\ | \end{bmatrix} \otimes \begin{bmatrix} | \\ \mathbf{x} \\ | \end{bmatrix} = \begin{bmatrix} a_0 & a_{n-1} & \cdots & a_1 \\ a_1 & a_0 & \cdots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

- $S = \{\mathbf{x} \in R \ : \ \|\mathbf{x}\|_\infty$ is small$\}$. (Note: $|S|$ is exponential in n.)

Theorem

"decoding" in cyclic lattices hard to approx in the worst case

$$\Downarrow$$

$f_{\mathbf{A}}$ *one-way on the average (for any width* $m = \omega(1)$*).*

Micciancio's Function

- $R = (\mathbb{Z}_p^n, +, \otimes)$, where $\otimes$ is cyclic convolution:

$$\begin{bmatrix} | \\ \mathbf{a} \\ | \end{bmatrix} \otimes \begin{bmatrix} | \\ \mathbf{x} \\ | \end{bmatrix} = \begin{bmatrix} a_0 & a_{n-1} & \cdots & a_1 \\ a_1 & a_0 & \cdots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

- $S = \{\mathbf{x} \in R \; : \; \|\mathbf{x}\|_\infty$ is small$\}$. (Note: $|S|$ is exponential in n.)

Theorem

"decoding" in cyclic lattices hard to approx in the worst case

$$\Downarrow$$

$f_{\mathbf{A}}$ one-way on the average (for any width $m = \omega(1)$).

Efficient: just m FFTs; small key

Micciancio's Function

- $R = (\mathbb{Z}_p^n, +, \otimes)$, where $\otimes$ is cyclic convolution:

$$\begin{bmatrix} | \\ \mathbf{a} \\ | \end{bmatrix} \otimes \begin{bmatrix} | \\ \mathbf{x} \\ | \end{bmatrix} = \begin{bmatrix} a_0 & a_{n-1} & \cdots & a_1 \\ a_1 & a_0 & \cdots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

- $S = \{\mathbf{x} \in R \ : \ \|\mathbf{x}\|_\infty$ is small$\}$. (Note: $|S|$ is exponential in n.)

Theorem

"decoding" in cyclic lattices hard to approx in the worst case

$$\Downarrow$$

$f_{\mathbf{A}}$ one-way on the average (for any width $m = \omega(1)$).

Efficient: just m FFTs; small key

Open Question: Like [Ajt96], is f collision-resistant?

Micciancio's Function

- $R = (\mathbb{Z}_p^n, +, \otimes)$, where $\otimes$ is cyclic convolution:

$$\begin{bmatrix} | \\ \mathbf{a} \\ | \end{bmatrix} \otimes \begin{bmatrix} | \\ \mathbf{x} \\ | \end{bmatrix} = \begin{bmatrix} a_0 & a_{n-1} & \cdots & a_1 \\ a_1 & a_0 & \cdots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

- $S = \{\mathbf{x} \in R \ : \ \|\mathbf{x}\|_\infty$ is small$\}$. (Note: $|S|$ is exponential in n.)

Theorem

"decoding" in cyclic lattices hard to approx in the worst case

$$\Downarrow$$

$f_{\mathbf{A}}$ one-way on the average (for any width $m = \omega(1)$).

Efficient: just m FFTs; small key

Open Question: Like [Ajt96], is f collision-resistant?

Today: *No!* (But we have a remedy...)

Collisions via an Algebraic View

Ring $R = \mathbb{Z}_p^n$ under $\otimes$ has algebraic structure:

$$\mathbf{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{Z}_p^n \quad \leftrightarrows \quad \mathbf{x}(\alpha) = \sum x_j \alpha^j \in \mathbb{Z}_p[\alpha]$$

Collisions via an Algebraic View

Ring $R = \mathbb{Z}_p^n$ under $\otimes$ has algebraic structure:

$$\mathbf{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{Z}_p^n \quad \leftrightarrows \quad \mathbf{x}(\alpha) = \sum x_j \alpha^j \in \mathbb{Z}_p[\alpha]$$

Fact 1: Convolution is polynomial multiplication, mod $\alpha^n - 1$.

$$\mathbf{a} \otimes \mathbf{x} \quad \leftrightarrows \quad \mathbf{a}(\alpha) \cdot \mathbf{x}(\alpha) \bmod (\alpha^n - 1)$$

Collisions via an Algebraic View

Ring $R = \mathbb{Z}_p^n$ under $\otimes$ has algebraic structure:

$$\mathbf{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{Z}_p^n \quad \leftrightarrows \quad \mathbf{x}(\alpha) = \sum x_j \alpha^j \in \mathbb{Z}_p[\alpha]$$

Fact 1: Convolution is polynomial multiplication, mod $\alpha^n - 1$.

$$\mathbf{a} \otimes \mathbf{x} \quad \leftrightarrows \quad \mathbf{a}(\alpha) \cdot \mathbf{x}(\alpha) \bmod (\alpha^n - 1)$$

Fact 2: Modulus $\alpha^n - 1$ is reducible.
$(\alpha^n - 1) = (\alpha - 1)(\alpha^{n-1} + \cdots + 1)$

Collisions via an Algebraic View

Ring $R = \mathbb{Z}_p^n$ under $\otimes$ has algebraic structure:

$$\mathbf{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{Z}_p^n \quad \leftrightarrows \quad \mathbf{x}(\alpha) = \sum x_j \alpha^j \in \mathbb{Z}_p[\alpha]$$

Fact 1: Convolution is polynomial multiplication, mod $\alpha^n - 1$.

$$\mathbf{a} \otimes \mathbf{x} \quad \leftrightarrows \quad \mathbf{a}(\alpha) \cdot \mathbf{x}(\alpha) \bmod (\alpha^n - 1)$$

Fact 2: Modulus $\alpha^n - 1$ is reducible.
$(\alpha^n - 1) = (\alpha - 1)(\alpha^{n-1} + \cdots + 1)$

Fact 3: $(\alpha - 1)$ divides uniform $\mathbf{a}_i(\alpha)$ in $\mathbb{Z}_p[\alpha]$ w/prob $1/p$.

Collisions via an Algebraic View

Ring $R = \mathbb{Z}_p^n$ under $\otimes$ has algebraic structure:

$$\mathbf{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{Z}_p^n \quad \leftrightarrows \quad \mathbf{x}(\alpha) = \sum x_j \alpha^j \in \mathbb{Z}_p[\alpha]$$

Fact 1: Convolution is polynomial multiplication, mod $\alpha^n - 1$.

$$\mathbf{a} \otimes \mathbf{x} \quad \leftrightarrows \quad \mathbf{a}(\alpha) \cdot \mathbf{x}(\alpha) \bmod (\alpha^n - 1)$$

Fact 2: Modulus $\alpha^n - 1$ is reducible.
$(\alpha^n - 1) = (\alpha - 1)(\alpha^{n-1} + \cdots + 1)$

Fact 3: $(\alpha - 1)$ divides uniform $\mathbf{a}_i(\alpha)$ in $\mathbb{Z}_p[\alpha]$ w/prob $1/p$.

Yields a collision:

$$\mathbf{a}_i(\alpha) \cdot \underbrace{(\alpha^{n-1} + \cdots + 1)}_{\mathbf{x}_i} \quad = \quad \mathbf{a}_i(\alpha) \cdot \underbrace{0}_{\mathbf{x}_i'} \quad \bmod (\alpha^n - 1)$$

Works because $\mathbb{Z}_p[\alpha]/(\alpha^n - 1)$ is *not* an integral domain.

Our Function

Choose n prime.

- $(\alpha - 1)$ and $(\alpha^{n-1} + \cdots + 1)$ are irreducible in $\mathbb{Z}[\alpha]$.
- So arithmetic $\bmod(\alpha^n - 1)$ decomposes into *two* integral domains. (Chinese remaindering)

Our Function

Choose n prime.

- $(\alpha - 1)$ and $(\alpha^{n-1} + \cdots + 1)$ are irreducible in $\mathbb{Z}[\alpha]$.
- So arithmetic $\bmod(\alpha^n - 1)$ decomposes into *two* integral domains. (Chinese remaindering)

Then:

☞ $R = (\mathbb{Z}_p^{\mathbf{n}}, +, \otimes)$

☞ $S = \{\mathbf{x} \in R \ : \ \|\mathbf{x}\|_\infty$ small, and $(\alpha - 1) \mid \mathbf{x}(\alpha)$ in $\mathbb{Z}[\alpha]\}$.

Our Function

Choose n prime.

- $(\alpha - 1)$ and $(\alpha^{n-1} + \cdots + 1)$ are irreducible in $\mathbb{Z}[\alpha]$.
- So arithmetic $\bmod (\alpha^n - 1)$ decomposes into *two* integral domains. (Chinese remaindering)

Then:

☞ $R = (\mathbb{Z}_p^{\mathbf{n}}, +, \otimes)$

☞ $S = \{\mathbf{x} \in R \ : \ \|\mathbf{x}\|_\infty$ small, and $(\alpha - 1) \mid \mathbf{x}(\alpha)$ in $\mathbb{Z}[\alpha]\}$.

☞ Rules out our collisions, but is it provably secure?

Our Function

Choose n prime.

- $(\alpha - 1)$ and $(\alpha^{n-1} + \cdots + 1)$ are irreducible in $\mathbb{Z}[\alpha]$.
- So arithmetic $\bmod(\alpha^n - 1)$ decomposes into *two* integral domains. (Chinese remaindering)

Then:

- ☞ $R = (\mathbb{Z}_p^{\mathbf{n}}, +, \otimes)$
- ☞ $S = \{\mathbf{x} \in R \ : \ \|\mathbf{x}\|_\infty$ small, and $(\alpha - 1) \mid \mathbf{x}(\alpha)$ in $\mathbb{Z}[\alpha]\}$.
- ☞ Rules out our collisions, but is it provably secure?

Theorem (Us)

shortest vec in cyclic lattices hard to approx in worst case (prime n)

$$\Downarrow$$

$f_{\mathbf{A}}$ collision-resistant on the average, for width $m = O(1)$!

Our Function

Choose n prime.

- $(\alpha - 1)$ and $(\alpha^{n-1} + \cdots + 1)$ are irreducible in $\mathbb{Z}[\alpha]$.
- So arithmetic $\bmod(\alpha^n - 1)$ decomposes into *two* integral domains. (Chinese remaindering)

Then:

- ☞ $R = (\mathbb{Z}_p^{\mathbf{n}}, +, \otimes)$
- ☞ $S = \{\mathbf{x} \in R \;:\; \|\mathbf{x}\|_\infty$ small, and $(\alpha - 1) \mid \mathbf{x}(\alpha)$ in $\mathbb{Z}[\alpha]\}$.
- ☞ Rules out our collisions, but is it provably secure?

Theorem (Us)

shortest vec in cyclic lattices hard to approx in worst case (prime n)

$$\Downarrow$$

$f_{\mathbf{A}}$ collision-resistant on the average, for width $m = O(1)$!

$\boxed{\textit{Very efficient: even 2 FFTs suffice}}$

Our Function

Choose n prime.

- $(\alpha - 1)$ and $(\alpha^{n-1} + \cdots + 1)$ are irreducible in $\mathbb{Z}[\alpha]$.
- So arithmetic $\bmod(\alpha^n - 1)$ decomposes into *two* integral domains. (Chinese remaindering)

Then:

- ☞ $R = (\mathbb{Z}_p^{\mathbf{n}}, +, \otimes)$
- ☞ $S = \{\mathbf{x} \in R \ : \ \|\mathbf{x}\|_\infty$ small, and $(\alpha - 1) \mid \mathbf{x}(\alpha)$ in $\mathbb{Z}[\alpha]\}$.
- ☞ Rules out our collisions, but is it provably secure?

Theorem (Us, LM)

shortest vec in cyclic lattices hard to approx in worst case (prime n)

$$\Downarrow$$

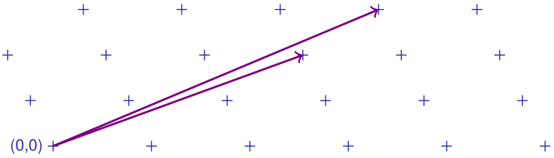
$f_{\mathbf{A}}$ *collision-resistant on the average, for width* $m = O(1)$*!*

$\boxed{\textit{Very efficient: even 2 FFTs suffice}}$

(Cyclic) Lattices

Let $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{Z}^n$ be linearly independent.
The lattice $\mathcal{L}(\mathbf{B}) \subset \mathbb{Z}^n$ having basis $\mathbf{B}$ is:

$$\mathcal{L}(\mathbf{B}) = \left\{ \sum_{i=1}^{d} c_i \mathbf{b}_i \mid \forall\, i, c_i \in \mathbb{Z} \right\}.$$

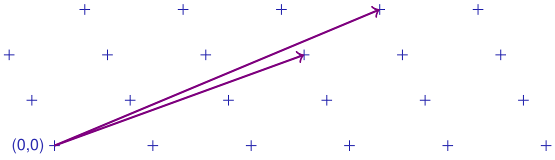


(Cyclic) Lattices

Let $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{Z}^n$ be linearly independent.
The lattice $\mathcal{L}(\mathbf{B}) \subset \mathbb{Z}^n$ having basis $\mathbf{B}$ is:

$$\mathcal{L}(\mathbf{B}) = \left\{ \sum_{i=1}^{d} c_i \mathbf{b}_i \mid \forall\, i, c_i \in \mathbb{Z} \right\}.$$

Lattice Λ is cyclic if $\mathbf{x} \in \Lambda \Rightarrow \mathrm{rot}(\mathbf{x}) \in \Lambda$.
For $\mathbf{x} = (x_0, \ldots, x_{n-1})$: $\quad \mathrm{rot}(\mathbf{x}) = (x_{n-1}, x_0, \ldots, x_{n-2})$.



(Cyclic) Lattices

Let $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{Z}^n$ be linearly independent.
The lattice $\mathcal{L}(\mathbf{B}) \subset \mathbb{Z}^n$ having basis $\mathbf{B}$ is:

$$\mathcal{L}(\mathbf{B}) = \left\{ \sum_{i=1}^{d} c_i \mathbf{b}_i \mid \forall\, i, c_i \in \mathbb{Z} \right\}.$$

Lattice Λ is cyclic if $\mathbf{x} \in \Lambda \Rightarrow \mathrm{rot}(\mathbf{x}) \in \Lambda$.
For $\mathbf{x} = (x_0, \ldots, x_{n-1})$: $\quad \mathrm{rot}(\mathbf{x}) = (x_{n-1}, x_0, \ldots, x_{n-2})$.

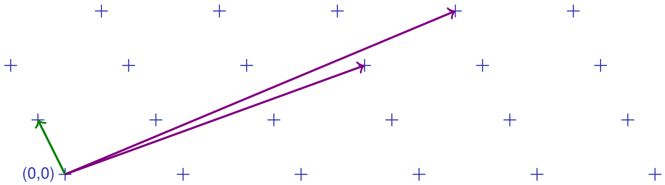
Cyclic lattices are closed under convolution with any $\mathbf{v} \in \mathbb{Z}^n$:

$$\mathbf{x} \otimes \mathbf{v} = \begin{pmatrix} x_0 & x_{n-1} & \cdots & x_1 \\ x_1 & x_0 & \cdots & x_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-1} & x_{n-2} & \cdots & x_0 \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{pmatrix} \in \Lambda.$$

Complexity of Shortest Vector

Shortest Vector Problem (SVP)

Given $\mathbf{B}$, find $\mathbf{v} \in \mathcal{L}(\mathbf{B}), \mathbf{v} \neq 0$ s.t. $\|\mathbf{v}\|$ (approx) minimal.



Complexity of Shortest Vector

Shortest Vector Problem (SVP)

Given $\mathbf{B}$, find $\mathbf{v} \in \mathcal{L}(\mathbf{B}), \mathbf{v} \neq 0$ s.t. $\|\mathbf{v}\|$ (approx) minimal.

Complexity

- In general, NP-hard to approx to any const fact [Ajt, Mic, Kho]. *But no NP-hardness known for cyclic lattices.*

Complexity of Shortest Vector

Shortest Vector Problem (SVP)

Given $\mathbf{B}$, find $\mathbf{v} \in \mathcal{L}(\mathbf{B}), \mathbf{v} \neq 0$ s.t. $\|\mathbf{v}\|$ (approx) minimal.

Complexity

- In general, NP-hard to approx to any const fact [Ajt, Mic, Kho].
  But no NP-hardness known for cyclic lattices.
- Best (general) algorithms yield approx factors $2^{\tilde{\Theta}(n)}$ [LLL, Sch].
  Don't seem to perform better on cyclic lattices.

Shortest Vector Problem (SVP)

Given $\mathbf{B}$, find $\mathbf{v} \in \mathcal{L}(\mathbf{B}), \mathbf{v} \neq 0$ s.t. $\|\mathbf{v}\|$ (approx) minimal.

Complexity

- In general, NP-hard to approx to any const fact [Ajt, Mic, Kho].
  But no NP-hardness known for cyclic lattices.
- Best (general) algorithms yield approx factors $2^{\tilde{\Theta}(n)}$ [LLL, Sch].
  Don't seem to perform better on cyclic lattices.

(We can't solve it, either!)

Complexity of Shortest Vector

Shortest Vector Problem (SVP)

Given $\mathbf{B}$, find $\mathbf{v} \in \mathcal{L}(\mathbf{B}), \mathbf{v} \neq 0$ s.t. $\|\mathbf{v}\|$ (approx) minimal.

Complexity

- In general, NP-hard to approx to any const fact [Ajt, Mic, Kho].
  But no NP-hardness known for cyclic lattices.
- Best (general) algorithms yield approx factors $2^{\tilde{\Theta}(n)}$ [LLL, Sch].
  Don't seem to perform better on cyclic lattices.

(We can't solve it, either!)

Our Assumption

For prime dimensions n, SVP hard to approx
to within $\tilde{\Theta}(n)$ in *cyclic* lattices, *in the worst case*.

Our New Understanding of Cyclic Lattices

- ☞ Linear algebra of cyclic lattices is tied to polynomial algebra.

Our New Understanding of Cyclic Lattices

☞ Linear algebra of cyclic lattices is tied to polynomial algebra.

For any polynomial $\Phi(\alpha) \mid (\alpha^n - 1)$, define the linear subspace:

$$H_\Phi = \{\mathbf{x} \in \mathbb{R}^n \ : \ \Phi(\alpha) \text{ divides } \mathbf{x}(\alpha) \text{ in } \mathbb{R}[\alpha]\}$$

Our New Understanding of Cyclic Lattices

☞ Linear algebra of cyclic lattices is tied to polynomial algebra.

For any polynomial $\Phi(\alpha) \mid (\alpha^n - 1)$, define the linear subspace:

$$H_\Phi = \{\mathbf{x} \in \mathbb{R}^n \ : \ \Phi(\alpha) \text{ divides } \mathbf{x}(\alpha) \text{ in } \mathbb{R}[\alpha]\}$$

Lemma 1: H_Φ is closed under rot (cyclic shift).

Our New Understanding of Cyclic Lattices

☞ Linear algebra of cyclic lattices is tied to polynomial algebra.

For any polynomial $\Phi(\alpha) \mid (\alpha^n - 1)$, define the linear subspace:

$$H_\Phi = \{\mathbf{x} \in \mathbb{R}^n \ : \ \Phi(\alpha) \text{ divides } \mathbf{x}(\alpha) \text{ in } \mathbb{R}[\alpha]\}$$

Lemma 1: H_Φ is closed under rot (cyclic shift).

Lemma 2: Let n be prime, and $\mathbf{x} \in \Lambda \cap H_{\alpha-1}$. Then

$$\mathbf{x}, \ \mathrm{rot}(\mathbf{x}), \ \ldots, \ \mathrm{rot}^{n-2}(\mathbf{x})$$

are linearly independent, and span $H_{\alpha-1}$.

Our New Understanding of Cyclic Lattices

☞ Linear algebra of cyclic lattices is tied to polynomial algebra.

For any polynomial $\Phi(\alpha) \mid (\alpha^n - 1)$, define the linear subspace:

$$H_\Phi = \{\mathbf{x} \in \mathbb{R}^n \ : \ \Phi(\alpha) \text{ divides } \mathbf{x}(\alpha) \text{ in } \mathbb{R}[\alpha]\}$$

Lemma 1: H_Φ is closed under rot (cyclic shift).

Lemma 2: Let n be prime, and $\mathbf{x} \in \Lambda \cap H_{\alpha-1}$. Then

$$\mathbf{x}, \ \mathrm{rot}(\mathbf{x}), \ \ldots, \ \mathrm{rot}^{n-2}(\mathbf{x})$$

are linearly independent, and span $H_{\alpha-1}$.

Lemma 3: shortest in $\Lambda \approx$ shortest in $(\Lambda \cap H_{\alpha-1})$.

Our New Understanding of Cyclic Lattices

☞ Linear algebra of cyclic lattices is tied to polynomial algebra.

For any polynomial $\Phi(\alpha) \mid (\alpha^n - 1)$, define the linear subspace:

$$H_\Phi = \{\mathbf{x} \in \mathbb{R}^n \ : \ \Phi(\alpha) \text{ divides } \mathbf{x}(\alpha) \text{ in } \mathbb{R}[\alpha]\}$$

Lemma 1: H_Φ is closed under rot (cyclic shift).

Lemma 2: Let n be prime, and $\mathbf{x} \in \Lambda \cap H_{\alpha-1}$. Then

$$\mathbf{x},\ \mathrm{rot}(\mathbf{x}),\ \ldots,\ \mathrm{rot}^{n-2}(\mathbf{x})$$

are linearly independent, and span $H_{\alpha-1}$.

Lemma 3: shortest in $\Lambda \approx$ shortest in $(\Lambda \cap H_{\alpha-1})$.

Corollary: $H_{\alpha-1}$ is "hard-core" for SVP.

Worst-Case to Average-Case Reduction

Solve SVP in $H_{\alpha-1}$

For *any* $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{Z}^n$ generating lattice Λ,
approximate shortest $\mathbf{v} \in \Lambda \cap H_{\alpha-1}$.

Worst-Case to Average-Case Reduction

Solve SVP in $H_{\alpha-1}$

For *any* $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{Z}^n$ generating lattice Λ,
approximate shortest $\mathbf{v} \in \Lambda \cap H_{\alpha-1}$.

Given

Oracle $\mathcal{O}$ finds collisions in our $f_{\mathbf{A}}$, but only for uniform keys $\mathbf{A}$.

Worst-Case to Average-Case Reduction

Solve SVP in $H_{\alpha-1}$

For *any* $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{Z}^n$ generating lattice Λ,
approximate shortest $\mathbf{v} \in \Lambda \cap H_{\alpha-1}$.

Given

Oracle $\mathcal{O}$ finds collisions in our $f_{\mathbf{A}}$, but only for uniform keys $\mathbf{A}$.

Reduction

Resembles [Ajt96, GGH97, CN97, M02, M'02, MR04], with improvements:

- ✔ "Bad" oracle answers are very rare (with elementary proof). (Integral domain.)
- ✔ Each iteration needs to find *only one* vector (not n). (Rotations are lin indep.)
- $\Rightarrow$ Simpler, tighter security reduction.

Worst-Case to Average-Case Reduction

Solve SVP in $H_{\alpha-1}$

For *any* $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{Z}^n$ generating lattice Λ,
approximate shortest $\mathbf{v} \in \Lambda \cap H_{\alpha-1}$.

Given

Oracle $\mathcal{O}$ finds collisions in our $f_{\mathbf{A}}$, but only for uniform keys $\mathbf{A}$.

Reduction

Resembles [Ajt96, GGH97, CN97, M02, M'02, MR04], with improvements:

- ✔ "Bad" oracle answers are very rare (with elementary proof). (Integral domain.)
- ✔ Each iteration needs to find *only one* vector (not n). (Rotations are lin indep.)

$\Rightarrow$ Simpler, tighter security reduction.

Worst-Case to Average-Case Reduction

Solve SVP in $H_{\alpha-1}$

For *any* $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{Z}^n$ generating lattice Λ, approximate shortest $\mathbf{v} \in \Lambda \cap H_{\alpha-1}$.

Given

Oracle $\mathcal{O}$ finds collisions in our $f_{\mathbf{A}}$, but only for uniform keys $\mathbf{A}$.

Reduction

Resembles [Ajt96, GGH97, CN97, M02, M'02, MR04], with improvements:

- ✔ "Bad" oracle answers are very rare (with elementary proof). (Integral domain.)
- ✔ Each iteration needs to find *only one* vector (not n). (Rotations are lin indep.)
- $\Rightarrow$ Simpler, tighter security reduction.

Conclusions

- ☞ Cyclic lattices yield very efficient cryptographic functions.

Conclusions

☞ Cyclic lattices yield very efficient cryptographic functions.

- More algebraic structure than general lattices.
- Tightly-connected computational problems.

Conclusions

☞ Cyclic lattices yield very efficient cryptographic functions.

- More algebraic structure than general lattices.
- Tightly-connected computational problems.

Open Question

What is their worst-case complexity?

Conclusions

☞ Cyclic lattices yield very efficient cryptographic functions.

- More algebraic structure than general lattices.
- Tightly-connected computational problems.

Open Question

What is their worst-case complexity?