A New Learning Problem with Applications To Cryptography

William Skeith

CCNY and Graduate Center
CAISS

Joint work with Gilbert Baumslag, Nelly Fazio, Antonio Nicolosi and Vladimir Shpilrain

Outline

1. **Motivation & Background**
   - Why Group-Theoretic Cryptography?
   - Learning With Errors (LWE)
2. **Generalized Learning Problems**
   - An Abstract Learning Problem
   - The Search for Instantiations: $B(n, 3)$
3. **Symmetric-Key Cryptosystem**
   - High-Level Approach
   - Construction

Outline

1. **Motivation & Background**
   - Why Group-Theoretic Cryptography?
   - Learning With Errors (LWE)
2. **Generalized Learning Problems**
   - An Abstract Learning Problem
   - The Search for Instantiations: $B(n, 3)$
3. **Symmetric-Key Cryptosystem**
   - High-Level Approach
   - Construction

Motivation

- Interesting mathematical problem on its own . . .
- Tackling crypto challenges of post-quantum era [Sh'94]
  - Shor's algorithm: Efficient *quantum* procedure to compute the order of any element in a cyclic group
- Quantum computing aside, diversifying assumptions still seems prudent

Motivation

- Interesting mathematical problem on its own ...
- Tackling crypto challenges of post-quantum era [Sh'94]
  - Shor's algorithm: Efficient *quantum* procedure to compute the order of any element in a cyclic group
  - Hardness of order-finding at the heart of most popular public-key cryptosystems (RSA, DH, ECDH)
  - ∴ If quantum computing becomes practical, we'll need alternative crypto platforms
- Quantum computing aside, diversifying assumptions still seems prudent

Motivation

- Interesting mathematical problem on its own . . .
- Tackling crypto challenges of post-quantum era [Sh'94]
  - Shor's algorithm: Efficient *quantum* procedure to compute the order of any element in a cyclic group
  - Hardness of order-finding at the heart of most popular public-key cryptosystems (RSA, DH, ECDH)

  ∴ If quantum computing becomes practical, we'll need alternative crypto platforms
- Quantum computing aside, diversifying assumptions still seems prudent

Motivation

- Interesting mathematical problem on its own . . .
- Tackling crypto challenges of post-quantum era [Sh'94]
  - Shor's algorithm: Efficient *quantum* procedure to compute the order of any element in a cyclic group
  - Hardness of order-finding at the heart of most popular public-key cryptosystems (RSA, DH, ECDH)
  - $\therefore$ If quantum computing becomes practical, we'll need alternative crypto platforms
- Quantum computing aside, diversifying assumptions still seems prudent

Motivation

- Interesting mathematical problem on its own ...
- Tackling crypto challenges of post-quantum era [Sh'94]
  - Shor's algorithm: Efficient *quantum* procedure to compute the order of any element in a cyclic group
  - Hardness of order-finding at the heart of most popular public-key cryptosystems (RSA, DH, ECDH)

  $\therefore$ If quantum computing becomes practical, we'll need alternative crypto platforms
- Quantum computing aside, diversifying assumptions still seems prudent

Motivation

- Interesting mathematical problem on its own . . .
- Tackling crypto challenges of post-quantum era [Sh'94]
    - Shor's algorithm: Efficient *quantum* procedure to compute the order of any element in a cyclic group
    - Hardness of order-finding at the heart of most popular public-key cryptosystems (RSA, DH, ECDH)
    - $\therefore$ If quantum computing becomes practical, we'll need alternative crypto platforms
- Quantum computing aside, diversifying assumptions still seems prudent

Prior Work in Non-Commutative Cryptography

Challenging computational problems abound in group theory, however...

- Many hard problems are based on infinite groups
- This makes probabilistic modeling difficult
- Average-case hardness for many problems seems to be not well-understood

Prior Work in Non-Commutative Cryptography

Challenging computational problems abound in group theory, however...

- Many hard problems are based on infinite groups
- This makes probabilistic modeling difficult
- Average-case hardness for many problems seems to be not well-understood

Prior Work in Non-Commutative Cryptography

Challenging computational problems abound in group theory, however...

- Many hard problems are based on infinite groups
- This makes probabilistic modeling difficult
- Average-case hardness for many problems seems to be not well-understood

Prior Work in Non-Commutative Cryptography

Challenging computational problems abound in group theory, however...

- Many hard problems are based on infinite groups
- This makes probabilistic modeling difficult
- Average-case hardness for many problems seems to be not well-understood

In this Work

Goal

Inspired by the success of LWE and lattice-based cryptography, we seek a new source of viable intractability assumptions from learning problems in group theory.

Outline

1. **Motivation & Background**
   - Why Group-Theoretic Cryptography?
   - Learning With Errors (LWE)
2. **Generalized Learning Problems**
   - An Abstract Learning Problem
   - The Search for Instantiations: $B(n, 3)$
3. **Symmetric-Key Cryptosystem**
   - High-Level Approach
   - Construction

Learning With Errors [Reg05]

Let $\mathbf{s} \in \mathbb{F}_p^n$. The picture is as follows:

$$\begin{tikzcd} \mathbb{F}_p^n \arrow[d, "\mathbf{s} \cdot \_"'] & \ni & \mathbf{a} \arrow[d, mapsto, "\approx \mathbf{s} \cdot \mathbf{a}"] & & \\ \mathbb{F}_p & \ni & b & = & \mathbf{s} \cdot \mathbf{a} + e \end{tikzcd}$$

LWE, Informally

Roughly, the **Learning With Errors** problem is to recover $\mathbf{s}$ by sampling preimage-image pairs in the presence of some small "noise"

Learning With Errors [Reg05]

Let $\mathbf{s} \in \mathbb{F}_p^n$. The picture is as follows:

$$\begin{tikzcd} \mathbb{F}_p^n \arrow[d, "\mathbf{s}\cdot\_"'] & \ni & \mathbf{a} \arrow[d, mapsto, "\approx \mathbf{s}\cdot\mathbf{a}"] & & \\ \mathbb{F}_p & \ni & b & = & \mathbf{s}\cdot\mathbf{a}+e \end{tikzcd}$$

LWE, Informally

Roughly, the **Learning With Errors** problem is to recover $\mathbf{s}$ by sampling preimage-image pairs in the presence of some small "noise"

Let $\mathbf{s} \in \mathbb{F}_p^n$. The picture is as follows:

$$\begin{tikzcd}
\mathbb{F}_p^n \arrow[d, "\mathbf{s} \cdot _"'] & \ni & \mathbf{a} \arrow[d, mapsto, "\approx \mathbf{s} \cdot \mathbf{a}"] & & \\
\mathbb{F}_p & \ni & b & = & \mathbf{s} \cdot \mathbf{a} + e
\end{tikzcd}$$

LWE, Informally

Roughly, the **Learning With Errors** problem is to recover $\mathbf{s}$ by sampling preimage-image pairs in the presence of some small "noise"

Learning With Errors [Reg05]

More precisely, let

- $\mathbf{s} \in \mathbb{F}_p^n$
- Ψ be a discrete gaussian distribution over $\mathbb{F}_p$ centered at 0
- Define a distribution $\mathbf{A}_{\mathbf{s},\Psi}$ on $\mathbb{F}_p^n \times \mathbb{F}_p$ whose samples are pairs $(\mathbf{a}, b)$ where $\mathbf{a} \xleftarrow{\$} \mathbb{F}_p^n$, $b = \mathbf{s} \cdot \mathbf{a} + e$, $e \xleftarrow{\$} \Psi$

Definition

The **Learning With Errors** problem is to recover $\mathbf{s}$ by sampling the distribution $\mathbf{A}_{\mathbf{s},\Psi}$.

Learning With Errors [Reg05]

More precisely, let

- $\mathbf{s} \in \mathbb{F}_p^n$
- Ψ be a discrete gaussian distribution over $\mathbb{F}_p$ centered at 0
- Define a distribution $\mathbf{A}_{\mathbf{s},\Psi}$ on $\mathbb{F}_p^n \times \mathbb{F}_p$ whose samples are pairs $(\mathbf{a}, b)$ where $\mathbf{a} \xleftarrow{\$} \mathbb{F}_p^n$, $b = \mathbf{s} \cdot \mathbf{a} + e$, $e \xleftarrow{\$} \Psi$

Definition

The **Learning With Errors** problem is to recover $\mathbf{s}$ by sampling the distribution $\mathbf{A}_{\mathbf{s},\Psi}$.

More precisely, let

- $\mathbf{s} \in \mathbb{F}_p^n$
- Ψ be a discrete gaussian distribution over $\mathbb{F}_p$ centered at 0
- Define a distribution $\mathbf{A}_{\mathbf{s},\Psi}$ on $\mathbb{F}_p^n \times \mathbb{F}_p$ whose samples are pairs $(\mathbf{a}, b)$ where $\mathbf{a} \xleftarrow{\$} \mathbb{F}_p^n$, $b = \mathbf{s} \cdot \mathbf{a} + e$, $e \xleftarrow{\$} \Psi$

Definition

The **Learning With Errors** problem is to recover $\mathbf{s}$ by sampling the distribution $\mathbf{A}_{\mathbf{s},\Psi}$.

More precisely, let

- $\mathbf{s} \in \mathbb{F}_p^n$
- Ψ be a discrete gaussian distribution over $\mathbb{F}_p$ centered at 0
- Define a distribution $\mathbf{A}_{\mathbf{s},\Psi}$ on $\mathbb{F}_p^n \times \mathbb{F}_p$ whose samples are pairs $(\mathbf{a}, b)$ where $\mathbf{a} \xleftarrow{\$} \mathbb{F}_p^n$, $b = \mathbf{s} \cdot \mathbf{a} + e$, $e \xleftarrow{\$} \Psi$

Definition

The **Learning With Errors** problem is to recover $\mathbf{s}$ by sampling the distribution $\mathbf{A}_{\mathbf{s},\Psi}$.

Learning With Errors [Reg05]

More precisely, let

- $\mathbf{s} \in \mathbb{F}_p^n$
- Ψ be a discrete gaussian distribution over $\mathbb{F}_p$ centered at 0
- Define a distribution $\mathbf{A}_{\mathbf{s},\Psi}$ on $\mathbb{F}_p^n \times \mathbb{F}_p$ whose samples are pairs $(\mathbf{a}, b)$ where $\mathbf{a} \xleftarrow{\$} \mathbb{F}_p^n, b = \mathbf{s} \cdot \mathbf{a} + e, e \xleftarrow{\$} \Psi$

Definition

The **Learning With Errors** problem is to recover $\mathbf{s}$ by sampling the distribution $\mathbf{A}_{\mathbf{s},\Psi}$.

Hardness of LWE

- For noise parameters $> \sqrt{n}$ no sub-exponential algorithms are known
  - In fact, for this case reductions from **worst-case** lattice problems have been shown ([Reg05,Pei09])
- Very recently, [AuGe11] showed a sub-exponential algorithm for noise parameters $< \sqrt{n}$

Hardness of LWE

- For noise parameters $> \sqrt{n}$ no sub-exponential algorithms are known
  - In fact, for this case reductions from **worst-case** lattice problems have been shown ([Reg05,Pei09])
- Very recently, [AuGe11] showed a sub-exponential algorithm for noise parameters $< \sqrt{n}$

Hardness of LWE

- For noise parameters $> \sqrt{n}$ no sub-exponential algorithms are known
  - In fact, for this case reductions from **worst-case** lattice problems have been shown ([Reg05,Pei09])
- Very recently, [AuGe11] showed a sub-exponential algorithm for noise parameters $< \sqrt{n}$

Outline

1. Motivation & Background
   - Why Group-Theoretic Cryptography?
   - Learning With Errors (LWE)
2. **Generalized Learning Problems**
   - An Abstract Learning Problem
   - The Search for Instantiations: $B(n, 3)$
3. Symmetric-Key Cryptosystem
   - High-Level Approach
   - Construction

Learning Homomorphisms With Errors

Observation

LWE's formulation was mainly algebraic:

- Expressed in terms of homomorphisms
- Complexity reductions (worst case to average case, search to decision) also algebraic

This motivates the following

Question

Can similar learning problems yield viable intractability assumptions based on group theory?

Learning Homomorphisms With Errors

Observation

LWE's formulation was mainly algebraic:

- Expressed in terms of homomorphisms
- Complexity reductions (worst case to average case, search to decision) also algebraic

This motivates the following

Can similar learning problems yield viable intractability assumptions based on group theory?

Learning Homomorphisms With Errors

Observation

LWE's formulation was mainly algebraic:

- Expressed in terms of homomorphisms
- Complexity reductions (worst case to average case, search to decision) also algebraic

This motivates the following

Question

Can similar learning problems yield viable intractability assumptions based on group theory?

Learning Homomorphisms With Errors

Observation

LWE's formulation was mainly algebraic:

- Expressed in terms of homomorphisms
- Complexity reductions (worst case to average case, search to decision) also algebraic

This motivates the following

Question

Can similar learning problems yield viable intractability assumptions based on group theory?

Learning Homomorphisms With Errors

Observation

LWE's formulation was mainly algebraic:

- Expressed in terms of homomorphisms
- Complexity reductions (worst case to average case, search to decision) also algebraic

This motivates the following

Question

Can similar learning problems yield viable intractability assumptions based on group theory?

LWE Over Groups

Vector Spaces | **Groups**

$$\begin{tikzcd}
\mathbb{F}_p^n \arrow[d, "\mathbf{s}\cdot_"'] & \ni & \mathbf{a} \arrow[d, mapsto, "\approx \mathbf{s}\cdot\mathbf{a}"] \\
\mathbb{F}_p & \ni & b \\
 & & \| \\
 & & \mathbf{s}\cdot\mathbf{a}+e
\end{tikzcd}
\qquad
\begin{tikzcd}
G_n \arrow[d, "\varphi"'] & \ni & a \arrow[d, mapsto, "\approx \varphi(a)"] \\
P_n & \ni & b \\
 & & \| \\
 & & \varphi(a)e
\end{tikzcd}$$

Learning Homomorphisms from Images with Errors

Setup

- Let G_n and P_n be groups
- Set Γ_n, Ψ_n, distributions on G_n, P_n, resp.
- Let Φ_n be a distribution on the set of all homomorphisms, $\hom(G_n, P_n)$

The Distribution $\mathbf{A}_{\varphi,\Psi_n}$

For $\varphi \xleftarrow{\$} \Phi_n$, define the analogous distribution $\mathbf{A}_{\varphi,\Psi_n}$ on $G_n \times P_n$ whose samples are (a, b) where

- $a \xleftarrow{\$} \Gamma_n;$
- $e \xleftarrow{\$} \Psi_n;$
- $b = \varphi(a)e$

Learning Homomorphisms from Images with Errors

Setup

- Let G_n and P_n be groups
- Set Γ_n, Ψ_n, distributions on G_n, P_n, resp.
- Let Φ_n be a distribution on the set of all homomorphisms, $\hom(G_n, P_n)$

The Distribution $\mathbf{A}_{\varphi,\Psi_n}$

For $\varphi \xleftarrow{\$} \Phi_n$, define the analogous distribution $\mathbf{A}_{\varphi,\Psi_n}$ on $G_n \times P_n$ whose samples are (a, b) where

$$a \xleftarrow{\$} \Gamma_n;$$

$$e \xleftarrow{\$} \Psi_n;$$

$$b = \varphi(a)e$$

Learning Homomorphisms from Images with Errors

Setup

- Let G_n and P_n be groups
- Set Γ_n, Ψ_n, distributions on G_n, P_n, resp.
- Let Φ_n be a distribution on the set of all homomorphisms, $\hom(G_n, P_n)$

The Distribution $\mathbf{A}_{\varphi,\Psi_n}$

For $\varphi \xleftarrow{\$} \Phi_n$, define the analogous distribution $\mathbf{A}_{\varphi,\Psi_n}$ on $G_n \times P_n$ whose samples are (a, b) where

- $a \xleftarrow{\$} \Gamma_n$;
- $e \xleftarrow{\$} \Psi_n$;
- $b = \varphi(a)e$

Learning Homomorphisms from Images with Errors

Setup

- Let G_n and P_n be groups
- Set Γ_n, Ψ_n, distributions on G_n, P_n, resp.
- Let Φ_n be a distribution on the set of all homomorphisms, $\hom(G_n, P_n)$

The Distribution $\mathbf{A}_{\varphi,\Psi_n}$

For $\varphi \xleftarrow{\$} \Phi_n$, define the analogous distribution $\mathbf{A}_{\varphi,\Psi_n}$ on $G_n \times P_n$ whose samples are (a, b) where

- $a \xleftarrow{\$} \Gamma_n$;
- $e \xleftarrow{\$} \Psi_n$;
- $b = \varphi(a)e$

Learning Homomorphisms from Images with Errors

Setup

- Let G_n and P_n be groups
- Set Γ_n, Ψ_n, distributions on G_n, P_n, resp.
- Let Φ_n be a distribution on the set of all homomorphisms, $\hom(G_n, P_n)$

The Distribution $\mathbf{A}_{\varphi,\Psi_n}$

For $\varphi \xleftarrow{\$} \Phi_n$, define the analogous distribution $\mathbf{A}_{\varphi,\Psi_n}$ on $G_n \times P_n$ whose samples are (a, b) where

- $a \xleftarrow{\$} \Gamma_n$;
- $e \xleftarrow{\$} \Psi_n$;
- $b = \varphi(a)e$

Learning Homomorphisms from Images with Errors

Setup

- Let G_n and P_n be groups
- Set Γ_n, Ψ_n, distributions on G_n, P_n, resp.
- Let Φ_n be a distribution on the set of all homomorphisms, $\hom(G_n, P_n)$

The Distribution $\mathbf{A}_{\varphi,\Psi_n}$

For $\varphi \xleftarrow{\$} \Phi_n$, define the analogous distribution $\mathbf{A}_{\varphi,\Psi_n}$ on $G_n \times P_n$ whose samples are (a, b) where

- $a \xleftarrow{\$} \Gamma_n$;
- $e \xleftarrow{\$} \Psi_n$;
- $b = \varphi(a)e$

Learning Homomorphisms from Images with Errors

Setup

- Let G_n and P_n be groups
- Set Γ_n, Ψ_n, distributions on G_n, P_n, resp.
- Let Φ_n be a distribution on the set of all homomorphisms, $\hom(G_n, P_n)$

The Distribution $\mathbf{A}_{\varphi,\Psi_n}$

For $\varphi \xleftarrow{\$} \Phi_n$, define the analogous distribution $\mathbf{A}_{\varphi,\Psi_n}$ on $G_n \times P_n$ whose samples are (a, b) where

- $a \xleftarrow{\$} \Gamma_n$;
- $e \xleftarrow{\$} \Psi_n$;
- $b = \varphi(a)e$

Learning Homomorphisms from Images with Errors

Setup

- Let G_n and P_n be groups
- Set Γ_n, Ψ_n, distributions on G_n, P_n, resp.
- Let Φ_n be a distribution on the set of all homomorphisms, $\hom(G_n, P_n)$

The Distribution $\mathbf{A}_{\varphi,\Psi_n}$

For $\varphi \xleftarrow{\$} \Phi_n$, define the analogous distribution $\mathbf{A}_{\varphi,\Psi_n}$ on $G_n \times P_n$ whose samples are (a, b) where

- $a \xleftarrow{\$} \Gamma_n$;
- $e \xleftarrow{\$} \Psi_n$;
- $b = \varphi(a)e$

Learning Homomorphisms from Images with Errors

Search Problem

Given $\mathbf{A}_{\varphi,\Psi_n}$, recover φ.

Decision Problem

Given samples from an unknown distribution $\mathbf{R} \in \{\mathbf{A}_{\varphi,\Psi_n}, \mathbf{U}(G_n \times P_n)\}$, determine $\mathbf{R}$.

Hardness Assumption (Decision Version)

$$\mathbf{A}_{\varphi,\Psi_n} \underset{\mathrm{PPT}}{\approx} \mathbf{U}(G_n \times P_n)$$

Learning Homomorphisms from Images with Errors

Search Problem

Given $\mathbf{A}_{\varphi,\Psi_n}$, recover φ.

Decision Problem

Given samples from an unknown distribution $\mathbf{R} \in \{\mathbf{A}_{\varphi,\Psi_n}, \mathbf{U}(G_n \times P_n)\}$, determine $\mathbf{R}$.

Hardness Assumption (Decision Version)

$$\mathbf{A}_{\varphi,\Psi_n} \underset{\text{PPT}}{\approx} \mathbf{U}(G_n \times P_n)$$

Learning Homomorphisms from Images with Errors

Search Problem

Given $\mathbf{A}_{\varphi,\Psi_n}$, recover φ.

Decision Problem

Given samples from an unknown distribution $\mathbf{R} \in \{\mathbf{A}_{\varphi,\Psi_n}, \mathbf{U}(G_n \times P_n)\}$, determine $\mathbf{R}$.

Hardness Assumption (Decision Version)

$$\mathbf{A}_{\varphi,\Psi_n} \underset{\text{PPT}}{\approx} \mathbf{U}(G_n \times P_n)$$

Learning Homomorphisms from Images with Errors

Note that this is a proper generalization of the standard LWE problem, where

- $G_n := (\mathbb{F}_p^n, +)$ and $\Gamma_n := \mathbf{U}(\mathbb{F}_p^n)$
- $P_n := (\mathbb{F}_p, +)$ and $\Psi_n :=$ discrete gaussian
- $\varphi := \mathbf{s} \cdot \_$ and $\Phi_n := \mathbf{U}(\hom(\mathbb{F}_p^n, \mathbb{F}_p))$

Learning Homomorphisms from Images with Errors

Note that this is a proper generalization of the standard LWE problem, where

- $G_n := (\mathbb{F}_p^n, +)$ and $\Gamma_n := \mathbf{U}(\mathbb{F}_p^n)$
- $P_n := (\mathbb{F}_p, +)$ and $\Psi_n :=$ discrete gaussian
- $\varphi := \mathbf{s} \cdot \_$ and $\Phi_n := \mathbf{U}(\hom(\mathbb{F}_p^n, \mathbb{F}_p))$

Learning Homomorphisms from Images with Errors

Note that this is a proper generalization of the standard LWE problem, where

- $G_n := (\mathbb{F}_p^n, +)$ and $\Gamma_n := \mathbf{U}(\mathbb{F}_p^n)$
- $P_n := (\mathbb{F}_p, +)$ and $\Psi_n :=$ discrete gaussian
- $\varphi := \mathbf{s} \cdot \_$ and $\Phi_n := \mathbf{U}(\hom(\mathbb{F}_p^n, \mathbb{F}_p))$

Learning Homomorphisms from Images with Errors

Note that this is a proper generalization of the standard LWE problem, where

- $G_n := (\mathbb{F}_p^n, +)$ and $\Gamma_n := \mathbf{U}(\mathbb{F}_p^n)$
- $P_n := (\mathbb{F}_p, +)$ and $\Psi_n :=$ discrete gaussian
- $\varphi := \mathbf{s} \cdot \_$ and $\Phi_n := \mathbf{U}(\mathrm{hom}(\mathbb{F}_p^n, \mathbb{F}_p))$

Outline

1. Motivation & Background
   - Why Group-Theoretic Cryptography?
   - Learning With Errors (LWE)

2. **Generalized Learning Problems**
   - An Abstract Learning Problem
   - The Search for Instantiations: $B(n,3)$

3. Symmetric-Key Cryptosystem
   - High-Level Approach
   - Construction

Intuition for Hardness of LWE

Part of what makes LWE work is that $\mathbb{F}_p^n$ is a **free module**

Free Objects

- Any mapping of generators extends to a unique homomorphism
- Hence, space of keys is exponential in # generators
- Irrespective of the error distribution, $\varphi(a) + e$ always "looks" plausible as an image of a

So, what about free groups? Not such a good idea:

- Free groups are infinite—what to do about presentations/modding?
- Multiplication is transparent

Intuition for Hardness of LWE

Part of what makes LWE work is that $\mathbb{F}_p^n$ is a **free module**

Free Objects

- Any mapping of generators extends to a unique homomorphism
- Hence, space of keys is exponential in $\#$ generators
- Irrespective of the error distribution, $\varphi(a) + e$ always "looks" plausible as an image of a

So, what about free groups? Not such a good idea:

- Free groups are infinite—what to do about probabilistic modeling?
- Multiplication is transparent

Intuition for Hardness of LWE

Part of what makes LWE work is that $\mathbb{F}_p^n$ is a **free module**

Free Objects

- Any mapping of generators extends to a unique homomorphism
- Hence, space of keys is exponential in $\#$ generators
- Irrespective of the error distribution, $\varphi(a) + e$ always “looks” plausible as an image of a

So, what about free groups? Not such a good idea:

- Free groups are infinite—what to do about probabilistic modeling?
- Multiplication is transparent

Intuition for Hardness of LWE

Part of what makes LWE work is that $\mathbb{F}_p^n$ is a **free module**

Free Objects

- Any mapping of generators extends to a unique homomorphism
- Hence, space of keys is exponential in $\#$ generators
- Irrespective of the error distribution, $\varphi(a) + e$ always "looks" plausible as an image of a

So, what about free groups? Not such a good idea:

- Free groups are infinite—what to do about probabilistic modeling?
- Multiplication is transparent

Intuition for Hardness of LWE

Part of what makes LWE work is that $\mathbb{F}_p^n$ is a **free module**

Free Objects

- Any mapping of generators extends to a unique homomorphism
- Hence, space of keys is exponential in $\#$ generators
- Irrespective of the error distribution, $\varphi(a) + e$ always "looks" plausible as an image of a

So, what about free groups? Not such a good idea:

- Free groups are infinite—what to do about probabilistic modeling?
- Multiplication is transparent

Intuition for Hardness of LWE

Part of what makes LWE work is that $\mathbb{F}_p^n$ is a **free module**

Free Objects

- Any mapping of generators extends to a unique homomorphism
- Hence, space of keys is exponential in $\#$ generators
- Irrespective of the error distribution, $\varphi(a) + e$ always "looks" plausible as an image of a

So, what about free groups? Not such a good idea:

- Free groups are infinite—what to do about probabilistic modeling?
- Multiplication is transparent
  - Errors might be easy to separate
  - Subset sum is easy (no hope of a public key scheme using these techniques)
  - Length-based attacks? [MyUs07,HuTa00]

Intuition for Hardness of LWE

Part of what makes LWE work is that $\mathbb{F}_p^n$ is a **free module**

Free Objects

- Any mapping of generators extends to a unique homomorphism
- Hence, space of keys is exponential in $\#$ generators
- Irrespective of the error distribution, $\varphi(a) + e$ always "looks" plausible as an image of a

So, what about free groups? Not such a good idea:

- Free groups are infinite—what to do about probabilistic modeling?
- Multiplication is transparent
  - Errors might be easy to separate
  - Subset sum is easy (no hope of a public key scheme using these techniques)
  - Length-based attacks? [MyUs07,HuTa00]

Intuition for Hardness of LWE

Part of what makes LWE work is that $\mathbb{F}_p^n$ is a **free module**

Free Objects

- Any mapping of generators extends to a unique homomorphism
- Hence, space of keys is exponential in $\#$ generators
- Irrespective of the error distribution, $\varphi(a) + e$ always "looks" plausible as an image of a

So, what about free groups? Not such a good idea:

- Free groups are infinite—what to do about probabilistic modeling?
- Multiplication is transparent
  - Errors might be easy to separate
  - Subset sum is easy (no hope of a public key scheme using these techniques)
  - Length-based attacks? [MyUs07,HuTa00]

Intuition for Hardness of LWE

Part of what makes LWE work is that $\mathbb{F}_p^n$ is a **free module**

Free Objects

- Any mapping of generators extends to a unique homomorphism
- Hence, space of keys is exponential in $\#$ generators
- Irrespective of the error distribution, $\varphi(a) + e$ always "looks" plausible as an image of a

So, what about free groups? Not such a good idea:

- Free groups are infinite—what to do about probabilistic modeling?
- Multiplication is transparent
  - Errors might be easy to separate
  - Subset sum is easy (no hope of a public key scheme using these techniques)
  - Length-based attacks? [MyUs07,HuTa00]

Intuition for Hardness of LWE

Part of what makes LWE work is that $\mathbb{F}_p^n$ is a **free module**

Free Objects

- Any mapping of generators extends to a unique homomorphism
- Hence, space of keys is exponential in $\#$ generators
- Irrespective of the error distribution, $\varphi(a) + e$ always “looks” plausible as an image of a

So, what about free groups? Not such a good idea:

- Free groups are infinite—what to do about probabilistic modeling?
- Multiplication is transparent
  - Errors might be easy to separate
  - Subset sum is easy (no hope of a public key scheme using these techniques)
  - Length-based attacks? [MyUs07,HuTa00]

Intuition for Hardness of LWE

Part of what makes LWE work is that $\mathbb{F}_p^n$ is a **free module**

Free Objects

- Any mapping of generators extends to a unique homomorphism
- Hence, space of keys is exponential in $\#$ generators
- Irrespective of the error distribution, $\varphi(a) + e$ always "looks" plausible as an image of a

So, what about free groups? Not such a good idea:

- Free groups are infinite—what to do about probabilistic modeling?
- Multiplication is transparent
  - Errors might be easy to separate
  - Subset sum is easy (no hope of a public key scheme using these techniques)
  - Length-based attacks? [MyUs07,HuTa00]

Intuition for Hardness of LWE

- Free objects seem like the right approach, but free groups seem rather unsuitable (**infinite order**, *etc.*)
- However, in restricted classes of groups, one can find **finite free objects**
- $\mathbb{F}_p^n$ is actually an example, but we'll look for something more general / less constrained

Intuition for Hardness of LWE

- Free objects seem like the right approach, but free groups seem rather unsuitable (**infinite order**, *etc.*)
- However, in restricted classes of groups, one can find **finite free objects**
- $\mathbb{F}_p^n$ is actually an example, but we'll look for something more general / less constrained

Intuition for Hardness of LWE

- Free objects seem like the right approach, but free groups seem rather unsuitable (**infinite order**, *etc.*)
- However, in restricted classes of groups, one can find **finite free objects**
- $\mathbb{F}_p^n$ is actually an example, but we'll look for something more general / less constrained

The Search for Instantiations

$$\begin{tikzcd}
\text{Free groups} \arrow[d, dash] \\
\text{Other varieties} \arrow[d, dash] \\
\mathbb{F}_p^n
\end{tikzcd}$$

The Search for Instantiations

Free groups

Other varieties

$\mathbb{F}_p^n$ Too restricted (already studied)

The Search for Instantiations

Free groups — Too free?

Other varieties

$\mathbb{F}_p^n$ — Too restricted (already studied)

The Search for Instantiations

Free groups	Too free?
Other varieties	**We'll look here**
$\mathbb{F}_p^n$	Too restricted (already studied)

Variety of Groups (Informal)

Roughly speaking, a **variety** is the class of all groups whose elements satisfy a certain set of equations.

Example

Abelian groups can be seen as the variety corresponding to the equation

$$XY = YX.$$

Varieties of Groups

Variety of Groups (Informal)

Roughly speaking, a **variety** is the class of all groups whose elements satisfy a certain set of equations.

Example

Abelian groups can be seen as the variety corresponding to the equation

$$XY = YX.$$

Varieties of Groups

Via the usual “abstract nonsense”, it is easy to see that varieties of groups contain free objects—just take a free group and factor out the normal subgroup resulting from all the “equations”...

$$\begin{tikzcd}
S \arrow[r, "i"] \arrow[dr, "f"'] & UF(S) \arrow[d, dashed, "Uf'"] \\
 & UG
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
F(S) \arrow[r, "p"] \arrow[d, dashed, "f'"] & F(S)/E \arrow[dl, dashed, "f''"] \\
G &
\end{tikzcd}$$

Sets **Groups**

Varieties of Groups

Via the usual “abstract nonsense”, it is easy to see that varieties of groups contain free objects—just take a free group and factor out the normal subgroup resulting from all the “equations”...

$$
\begin{tikzcd}
S \arrow[r, "i"] \arrow[dr, "f"'] & UF(S) \arrow[d, dashed, "Uf'"] \\
 & UG
\end{tikzcd}
\qquad \Bigg| \qquad
\begin{tikzcd}
F(S) \arrow[r, "p"] \arrow[d, dashed, "f'"] & F(S)/E \arrow[dl, dashed, "f''"] \\
G &
\end{tikzcd}
$$

Sets **Groups**

Varieties of Groups

Question

Which varieties of groups contain **finite free objects**???

If the equations are say,

$$\begin{aligned} [X, Y] &= 1 \\ X^p &= 1 \end{aligned}$$

then the free objects are exactly $\mathbb{Z}_p^n$, which are the objects of study in LWE (if p is prime).

Question

What happens if the $[X, Y] = 1$ equation is removed?[a] In general, the answer is not so simple...

[a] Note: $[X, Y] = X^{-1}Y^{-1}XY$.

Varieties of Groups

Question

Which varieties of groups contain **finite free objects**???

If the equations are say,

$$\begin{aligned} [X, Y] &= 1 \\ X^p &= 1 \end{aligned}$$

then the free objects are exactly $\mathbb{Z}_p^n$, which are the objects of study in LWE (if p is prime).

Question

What happens if the $[X, Y] = 1$ equation is removed?[a] In general, the answer is not so simple...

[a] Note: $[X, Y] = X^{-1}Y^{-1}XY$.

Question

Which varieties of groups contain **finite free objects**???

If the equations are say,

$$\begin{aligned} [X, Y] &= 1 \\ X^p &= 1 \end{aligned}$$

then the free objects are exactly $\mathbb{Z}_p^n$, which are the objects of study in LWE (if p is prime).

Question

What happens if the $[X, Y] = 1$ equation is removed?[a] In general, the answer is not so simple...

[a] Note: $[X, Y] = X^{-1}Y^{-1}XY$.

Burnside Groups

Notation

For the variety of groups defined by the equation $X^m = 1$, denote the free group on n generators in this variety by $B(n, m)$.

Determining the finiteness of $B(n, m)$ is known as the **Bounded Burnside Problem**.

Burnside Groups

Notation

For the variety of groups defined by the equation $X^m = 1$, denote the free group on n generators in this variety by $B(n, m)$.

Determining the finiteness of $B(n, m)$ is known as the **Bounded Burnside Problem**.

Bounded Burnside Problem

For $n > 1$ and for sufficiently large m, it is known that $|B(n,m)| = \infty$, yet for small m, our understanding is far from complete:

$B(n,2)$	Finite (also abelian)
$B(n,3)$	Finite
$B(n,4)$	Finite
$B(n,5)$	**Unknown**
$B(n,6)$	Finite
$B(n,7)$	**Unknown**
$\vdots$	$\vdots$

Our Approach

We will use $B(n,3)$ as a starting point for our investigation: it is the simplest case yielding finiteness + non-abelian.

Normal Form for $B(n,3)$

The structure of $B(n,3)$ is fairly well-understood. In particular we have the following

Fact

Every element of $B(n,3)$ has a unique representation as

$$x_1^{\alpha_1} \cdots x_i^{\alpha_i} \cdots x_n^{\alpha_n} [x_1, x_2]^{\beta_{1,2}} \cdots [x_i, x_j]^{\beta_{i,j}} \cdots [x_{n-1}, x_n]^{\beta_{n-1,n}} [x_1, x_2, x_3]^{\gamma_{1,2,3}} \cdots [x_i, x_j, x_k]^{\gamma_{i,j,k}} \cdots [x_{n-2}, x_{n-1}, x_n]^{\gamma_{n-2,n-1,n}}$$

where the $\{x_i\}$ are the generators, all $\alpha_i, \beta_{i,j}, \gamma_{i,j,k} \in \{0,1,2\}$ for all $1 \le i < j < k \le n$, and $[x_i, x_j, x_k] = [[x_i, x_j], x_k]$.

Corollary

Given the above normal form, we see that the order of $B(n,3)$ is

$$3^{n + \binom{n}{2} + \binom{n}{3}}$$

Normal Form for $B(n,3)$

The structure of $B(n,3)$ is fairly well-understood. In particular we have the following

Fact

Every element of $B(n,3)$ has a unique representation as

$$x_1^{\alpha_1}\cdots x_i^{\alpha_i}\cdots x_n^{\alpha_n}[x_1,x_2]^{\beta_{1,2}}\cdots[x_i,x_j]^{\beta_{i,j}}\cdots[x_{n-1},x_n]^{\beta_{n-1,n}}[x_1,x_2,x_3]^{\gamma_{1,2,3}}\\ \cdots[x_i,x_j,x_k]^{\gamma_{i,j,k}}\cdots[x_{n-2},x_{n-1},x_n]^{\gamma_{n-2,n-1,n}}$$

where the $\{x_i\}$ are the generators, all $\alpha_i,\beta_{i,j},\gamma_{i,j,k}\in\{0,1,2\}$ for all $1\le i<j<k\le n$, and $[x_i,x_j,x_k]=[[x_i,x_j],x_k]$.

Corollary

Given the above normal form, we see that the order of $B(n,3)$ is

$$3^{n+\binom{n}{2}+\binom{n}{3}}$$

Normal Form for $B(n, 3)$

The structure of $B(n, 3)$ is fairly well-understood. In particular we have the following

Fact

Every element of $B(n, 3)$ has a unique representation as

$$x_1^{\alpha_1} \cdots x_i^{\alpha_i} \cdots x_n^{\alpha_n} [x_1, x_2]^{\beta_{1,2}} \cdots [x_i, x_j]^{\beta_{i,j}} \cdots [x_{n-1}, x_n]^{\beta_{n-1,n}} [x_1, x_2, x_3]^{\gamma_{1,2,3}} \cdots [x_i, x_j, x_k]^{\gamma_{i,j,k}} \cdots [x_{n-2}, x_{n-1}, x_n]^{\gamma_{n-2,n-1,n}}$$

where the $\{x_i\}$ are the generators, all $\alpha_i, \beta_{i,j}, \gamma_{i,j,k} \in \{0, 1, 2\}$ for all $1 \leq i < j < k \leq n$, and $[x_i, x_j, x_k] = [[x_i, x_j], x_k]$.

Corollary

Given the above normal form, we see that the order of $B(n, 3)$ is

$$3^{n + \binom{n}{2} + \binom{n}{3}}$$

Putting the Pieces Together...

Recall the setup:

$$G_n \xrightarrow{\varphi \xleftarrow{\$} \Phi_n} P_n$$

$$a \xleftarrow{\$} \Gamma_n \longmapsto \varphi(a)e, e \xleftarrow{\$} \Psi_n$$

Instantiating the Abstract Learning Problem

- G_n
- P_n
- Φ_n
- Γ_n
- Ψ_n

Putting the Pieces Together...

Recall the setup:

$$G_n \xrightarrow{\varphi \xleftarrow{\$} \Phi_n} P_n$$

$$a \xleftarrow{\$} \Gamma_n \longmapsto \varphi(a)e, e \xleftarrow{\$} \Psi_n$$

Instantiating the Abstract Learning Problem

- $G_n := B(n, 3)$
- P_n
- Φ_n
- Γ_n
- Ψ_n

Putting the Pieces Together...

Recall the setup:

$$G_n \xrightarrow{\varphi \xleftarrow{\$} \Phi_n} P_n$$

$$a \xleftarrow{\$} \Gamma_n \longmapsto \varphi(a)e,\ e \xleftarrow{\$} \Psi_n$$

Instantiating the Abstract Learning Problem

- $G_n := B(n, 3)$
- $P_n := B(r, 3)$, $r < n$
- Φ_n
- Γ_n
- Ψ_n

Putting the Pieces Together...

Recall the setup:

$$G_n \xrightarrow{\varphi \xleftarrow{\$} \Phi_n} P_n$$

$$a \xleftarrow{\$} \Gamma_n \longmapsto \varphi(a)e,\ e \xleftarrow{\$} \Psi_n$$

Instantiating the Abstract Learning Problem

- $G_n := B(n, 3)$
- $P_n := B(r, 3),\ r < n$
- $\Phi_n := \mathbf{U}(\hom(B(n, 3), B(r, 3)))$
- Γ_n
- Ψ_n

Putting the Pieces Together...

Recall the setup:

$$G_n \xrightarrow{\varphi \xleftarrow{\$} \Phi_n} P_n$$

$$a \xleftarrow{\$} \Gamma_n \longmapsto \varphi(a)e,\ e \xleftarrow{\$} \Psi_n$$

Instantiating the Abstract Learning Problem

- $G_n := B(n, 3)$
- $P_n := B(r, 3)$, $r < n$
- $\Phi_n := \mathbf{U}(\hom(B(n, 3), B(r, 3)))$ **Easy to sample:** $B(n, 3)$ **is free**
- Γ_n
- Ψ_n

Putting the Pieces Together...

Recall the setup:

$$\begin{tikzcd} G_n \arrow[r, "\varphi \xleftarrow{\$} \Phi_n"] & P_n \end{tikzcd}$$

$$a \xleftarrow{\$} \Gamma_n \longmapsto \varphi(a)e,\ e \xleftarrow{\$} \Psi_n$$

Instantiating the Abstract Learning Problem

- $G_n := B(n, 3)$
- $P_n := B(r, 3),\ r < n$
- $\Phi_n := \mathbf{U}(\hom(B(n, 3), B(r, 3)))$
- $\Gamma_n := \mathbf{U}(B(n, 3))$
- Ψ_n

Putting the Pieces Together...

Recall the setup:

$$G_n \xrightarrow{\varphi \xleftarrow{\$} \Phi_n} P_n$$

$$a \xleftarrow{\$} \Gamma_n \longmapsto \varphi(a)e,\ e \xleftarrow{\$} \Psi_n$$

Instantiating the Abstract Learning Problem

- $G_n := B(n, 3)$
- $P_n := B(r, 3)$, $r < n$
- $\Phi_n := \mathbf{U}(\hom(B(n, 3), B(r, 3)))$
- $\Gamma_n := \mathbf{U}(B(n, 3))$ **Easy to sample: *cf.* normal form**
- Ψ_n

Putting the Pieces Together...

Recall the setup:

$$\begin{tikzcd} G_n \arrow[rr, "\varphi \xleftarrow{\$} \Phi_n"] & & P_n \end{tikzcd}$$

$$a \xleftarrow{\$} \Gamma_n \longmapsto \varphi(a)e,\, e \xleftarrow{\$} \Psi_n$$

Instantiating the Abstract Learning Problem

- $G_n := B(n, 3)$
- $P_n := B(r, 3)$, $r < n$
- $\Phi_n := \mathbf{U}(\hom(B(n, 3), B(r, 3)))$
- $\Gamma_n := \mathbf{U}(B(n, 3))$
- $\Psi_n :=$ **???**

Putting the Pieces Together...

Recall the setup:

$$G_n \xrightarrow{\varphi \xleftarrow{\$} \Phi_n} P_n$$

$$a \xleftarrow{\$} \Gamma_n \longmapsto \varphi(a)e, e \xleftarrow{\$} \Psi_n$$

Instantiating the Abstract Learning Problem

- $G_n := B(n, 3)$
- $P_n := B(r, 3)$, $r < n$
- $\Phi_n := \mathbf{U}(\hom(B(n, 3), B(r, 3)))$
- $\Gamma_n := \mathbf{U}(B(n, 3))$
- $\Psi_n :=$ **???**

The error distribution requires more care...

Connection with LWE/$\mathbb{F}_3$

For certain error distributions, the decision problem over $B(n,3)$ would reduce to LWE with $p = 3$. Consider the abelianization:

$$\begin{tikzcd}
B(n,3) \arrow[r, "\varphi"] \arrow[d, "G \mapsto G/[G,G]"'] & B(r,3) \arrow[d] \\
\mathbb{F}_3^n \arrow[r, "\varphi'"'] & \mathbb{F}_3^r
\end{tikzcd}$$

This allows one to transform $\mathbf{A}_{\varphi,\Psi}$ over $B(n,3) \times B(r,3)$ to $\mathbf{A}_{\varphi',\Psi'}$ over $\mathbb{F}_3^n \times \mathbb{F}_3^r$ for some induced error distribution Ψ'. Hence the $B(n,3)$ LWE is no harder than the vector space LWE with the induced error Ψ'.

Connection with LWE/$\mathbb{F}_3$

For certain error distributions, the decision problem over $B(n,3)$ would reduce to LWE with $p = 3$. Consider the abelianization:

$$\begin{tikzcd}
B(n,3) \arrow[r, "\varphi"] \arrow[d, "G \mapsto G/[G,G]"'] & B(r,3) \arrow[d] \\
\mathbb{F}_3^n \arrow[r, "\varphi'"'] & \mathbb{F}_3^r
\end{tikzcd}$$

This allows one to transform $\mathbf{A}_{\varphi,\Psi}$ over $B(n,3) \times B(r,3)$ to $\mathbf{A}_{\varphi',\Psi'}$ over $\mathbb{F}_3^n \times \mathbb{F}_3^r$ for some induced error distribution Ψ'. Hence the $B(n,3)$ LWE is no harder than the vector space LWE with the induced error Ψ'.

Connection with LWE/$\mathbb{F}_3$

For certain error distributions, the decision problem over $B(n,3)$ would reduce to LWE with $p = 3$. Consider the abelianization:

$$\begin{tikzcd}
B(n,3) \arrow[rr, "\varphi"] \arrow[d, "G \mapsto G/[G,G]"'] & & B(r,3) \arrow[d] \\
\mathbb{F}_3^n \arrow[rr, "\varphi'"'] & & \mathbb{F}_3^r
\end{tikzcd}$$

This allows one to transform $\mathbf{A}_{\varphi,\Psi}$ over $B(n,3) \times B(r,3)$ to $\mathbf{A}_{\varphi',\Psi'}$ over $\mathbb{F}_3^n \times \mathbb{F}_3^r$ for some induced error distribution Ψ'. Hence the $B(n,3)$ LWE is no harder than the vector space LWE with the induced error Ψ'.

Error Distribution

In light of the preceding, we'll select an error distribution so that the abelianization construction takes $\mathbf{A}_{\varphi,\Psi}$ to **the uniform distribution** $\mathbf{U}(\mathbb{F}_3^n \times \mathbb{F}_3^r)$.

Ψ_n

Let $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_3^r$ and let $\sigma \xleftarrow{\$} S_r$ be a permutation. A sample from Ψ_n is an element

$$e = \prod_{i=1}^{r} x_{\sigma(i)}^{v_i}$$

where the $\{x_i\}$ are the generators of $B(r, 3)$ and the $\{v_i\}$ are the components of $\mathbf{v}$.

Moreover, notice that the normal closure of $\mathrm{Support}(\Psi)$ is in fact the entire group $B(r, 3)$. **Intuition:** this leaves no apparent way to "factor out" the noise.

Error Distribution

In light of the preceding, we'll select an error distribution so that the abelianization construction takes $\mathbf{A}_{\varphi,\Psi}$ to **the uniform distribution** $\mathbf{U}(\mathbb{F}_3^n \times \mathbb{F}_3^r)$.

Ψ_n

Let $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_3^r$ and let $\sigma \xleftarrow{\$} S_r$ be a permutation. A sample from Ψ_n is an element

$$e = \prod_{i=1}^{r} x_{\sigma(i)}^{v_i}$$

where the $\{x_i\}$ are the generators of $B(r,3)$ and the $\{v_i\}$ are the components of $\mathbf{v}$.

Moreover, notice that the normal closure of $\mathrm{Support}(\Psi)$ is in fact the entire group $B(r,3)$. **Intuition:** this leaves no apparent way to "factor out" the noise.

Error Distribution

In light of the preceding, we'll select an error distribution so that the abelianization construction takes $\mathbf{A}_{\varphi,\Psi}$ to **the uniform distribution** $\mathbf{U}(\mathbb{F}_3^n \times \mathbb{F}_3^r)$.

Ψ_n

Let $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_3^r$ and let $\sigma \xleftarrow{\$} S_r$ be a permutation. A sample from Ψ_n is an element

$$e = \prod_{i=1}^{r} x_{\sigma(i)}^{v_i}$$

where the $\{x_i\}$ are the generators of $B(r,3)$ and the $\{v_i\}$ are the components of $\mathbf{v}$.

Moreover, notice that the normal closure of $\mathrm{Support}(\Psi)$ is in fact the entire group $B(r,3)$. **Intuition:** this leaves no apparent way to "factor out" the noise.

Outline

1. Motivation & Background
   - Why Group-Theoretic Cryptography?
   - Learning With Errors (LWE)
2. Generalized Learning Problems
   - An Abstract Learning Problem
   - The Search for Instantiations: $B(n, 3)$
3. **Symmetric-Key Cryptosystem**
   - **High-Level Approach**
   - Construction

High-Level Approach

- Goal: construct a simple Regev-like cryptosystem which encrypts bits
- The secret key will be a homomorphism φ
- Encryptions of 0 will be noisy images of φ (*i.e.,* samples from $\mathbf{A}_{\varphi,\Psi}$)
- Encryptions of 1 will be "far" from a noisy image of φ

High-Level Approach

- Goal: construct a simple Regev-like cryptosystem which encrypts bits
- The secret key will be a homomorphism φ
- Encryptions of 0 will be noisy images of φ (*i.e.,* samples from $\mathbf{A}_{\varphi,\Psi}$)
- Encryptions of 1 will be "far" from a noisy image of φ

High-Level Approach

- Goal: construct a simple Regev-like cryptosystem which encrypts bits
- The secret key will be a homomorphism φ
- Encryptions of 0 will be noisy images of φ (*i.e.,* samples from $\mathbf{A}_{\varphi,\Psi}$)
- Encryptions of 1 will be "far" from a noisy image of φ

High-Level Approach

- Goal: construct a simple Regev-like cryptosystem which encrypts bits
- The secret key will be a homomorphism φ
- Encryptions of 0 will be noisy images of φ (*i.e.,* samples from $\mathbf{A}_{\varphi,\Psi}$)
- Encryptions of 1 will be "far" from a noisy image of φ

Additional Considerations for $B(n, 3)$

For this approach to make sense, we'll need a few more ingredients:

Required Ingredients

- Norm / distance metric on $B(r, 3)$
- "Large" diameter (must be able to distinguish noisy images from noise)

Additional Considerations for $B(n,3)$

For this approach to make sense, we'll need a few more ingredients:

Required Ingredients

- Norm / distance metric on $B(r,3)$
- "Large" diameter (must be able to distinguish noisy images from noise)

Additional Considerations for $B(n, 3)$

For this approach to make sense, we'll need a few more ingredients:

Required Ingredients

- Norm / distance metric on $B(r, 3)$
- "Large" diameter (must be able to distinguish noisy images from noise)

Additional Considerations for $B(n, 3)$

For this approach to make sense, we'll need a few more ingredients:

Required Ingredients

- Norm / distance metric on $B(r, 3)$
- "Large" diameter (must be able to distinguish noisy images from noise)

Cayley Graph

In response to our needs for a metric, we turn to the **Cayley Graph**.

Idea

- Treat a group as a geometric object
- Vertexes are elements; edges are generators (and their inverses)
- The **norm** (denoted $||g||$) is just the graph distance from the identity element

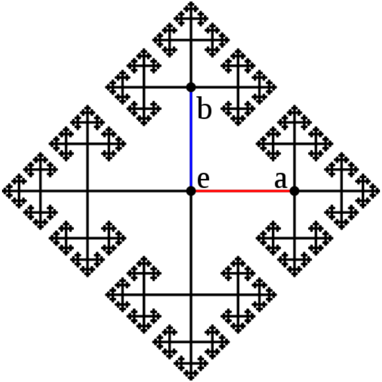

Figure: Cayley graph of $F(\{a, b\})$.

In response to our needs for a metric, we turn to the **Cayley Graph**.

Idea

- Treat a group as a geometric object
- Vertexes are elements; edges are generators (and their inverses)
- The **norm** (denoted $||g||$) is just the graph distance from the identity element

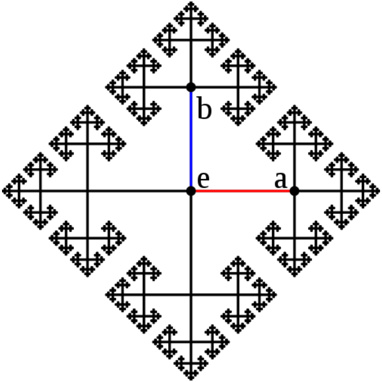

Figure: Cayley graph of $F(\{a, b\})$.

In response to our needs for a metric, we turn to the **Cayley Graph**.

Idea

- Treat a group as a geometric object
- Vertexes are elements; edges are generators (and their inverses)
- The **norm** (denoted $\|g\|$) is just the graph distance from the identity element

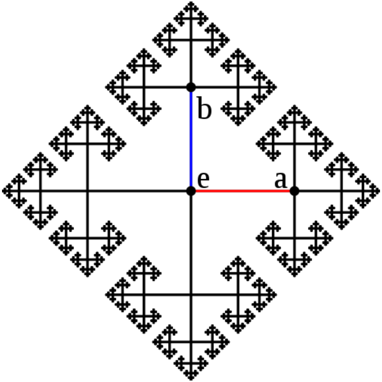

Figure: Cayley graph of $F(\{a, b\})$.

Diameter of $B(n,3)$

Just given the order of $B(n,3)$ alone, we can compute a simple lower bound on the diameter.

Lemma (Diameter of $B(n,3)$)

$\exists \tau_n \in B(n,3)$ *such that* $\|\tau_n\| \in \Omega(\frac{n^3}{\log n})$.

Diameter of $B(n, 3)$

Just given the order of $B(n, 3)$ alone, we can compute a simple lower bound on the diameter.

Lemma (Diameter of $B(n, 3)$)

$\exists \tau_n \in B(n, 3)$ *such that* $\|\tau_n\| \in \Omega(\frac{n^3}{\log n})$.

Diameter of $B(n, 3)$

Proof.

Let $d_n = \max_{x \in B(n,3)}(\|x\|)$, and recall that $|B(n,3)| = 3^{n+\binom{n}{2}+\binom{n}{3}}$. Since all elements of the group can be written with at most d_n symbols taken from $x_1^{\pm 1}, \ldots, x_n^{\pm 1}$:

$$
\begin{aligned}
(2n)^{d_n} &\geq 3^{n+\binom{n}{2}+\binom{n}{3}} \\
d_n \log_3(2n) &\geq n + \binom{n}{2} + \binom{n}{3} \\
d_n &\geq \left\lceil \frac{n + \binom{n}{2} + \binom{n}{3}}{\log_3 2n} \right\rceil && \text{(since } d_n \in \mathbb{Z}\text{)}
\end{aligned}
$$

Diameter of $B(n,3)$

Proof.

Let $d_n = \max_{x \in B(n,3)}(\|x\|)$, and recall that $|B(n,3)| = 3^{n+\binom{n}{2}+\binom{n}{3}}$. Since all elements of the group can be written with at most d_n symbols taken from $x_1^{\pm 1}, \ldots, x_n^{\pm 1}$:

$$
\begin{aligned}
(2n)^{d_n} &\geq 3^{n+\binom{n}{2}+\binom{n}{3}} \\
d_n \log_3(2n) &\geq n + \binom{n}{2} + \binom{n}{3} \\
d_n &\geq \left\lceil \frac{n + \binom{n}{2} + \binom{n}{3}}{\log_3 2n} \right\rceil \qquad \text{(since } d_n \in \mathbb{Z}\text{)}
\end{aligned}
$$

∎

Diameter of $B(n, 3)$

Good so far, but one issue remains: for a given $x \in G$, **how does one *compute* the norm** in the Cayley graph?

Computing Cayley Graph Norms

- In some cases, this is known to be NP-hard
- It wasn't until 2010 that the diameter of the Rubik's cube group was computed, and this took 35 CPU years...
- Efficient methods may exist for $B(n, 3)$, but we can get away with small values of n, and just use breadth-first search

Diameter of $B(n, 3)$

Good so far, but one issue remains: for a given $x \in G$, **how does one *compute* the norm** in the Cayley graph?

Computing Cayley Graph Norms

- In some cases, this is known to be NP-hard
- It wasn't until 2010 that the diameter of the Rubik's cube group was computed, and this took 35 CPU-years...
- Efficient methods may exist for $B(r, 3)$, but we can get away with small values of r, and just use breadth-first search

Diameter of $B(n, 3)$

Good so far, but one issue remains: for a given $x \in G$, **how does one *compute* the norm** in the Cayley graph?

Computing Cayley Graph Norms

- In some cases, this is known to be NP-hard
- It wasn't until 2010 that the diameter of the Rubik's cube group was computed, and this took 35 CPU-years...
- Efficient methods may exist for $B(r, 3)$, but we can get away with small values of r, and just use breadth-first search

Diameter of $B(n, 3)$

Good so far, but one issue remains: for a given $x \in G$, **how does one *compute* the norm** in the Cayley graph?

Computing Cayley Graph Norms

- In some cases, this is known to be NP-hard
- It wasn't until 2010 that the diameter of the Rubik's cube group was computed, and this took 35 CPU-years...
- Efficient methods may exist for $B(r, 3)$, but we can get away with small values of r, and just use breadth-first search

Diameter of $B(n, 3)$

Good so far, but one issue remains: for a given $x \in G$, **how does one *compute* the norm** in the Cayley graph?

Computing Cayley Graph Norms

- In some cases, this is known to be NP-hard
- It wasn't until 2010 that the diameter of the Rubik's cube group was computed, and this took 35 CPU-years...
- Efficient methods may exist for $B(r, 3)$, but we can get away with small values of r, and just use breadth-first search

Outline

1. **Motivation & Background**
   - Why Group-Theoretic Cryptography?
   - Learning With Errors (LWE)
2. **Generalized Learning Problems**
   - An Abstract Learning Problem
   - The Search for Instantiations: $B(n, 3)$
3. **Symmetric-Key Cryptosystem**
   - High-Level Approach
   - Construction

We can now proceed with a formal description of the cryptosystem.

Precomputation

Run breadth-first search on the Cayley graph of $B(r,3)$, recording the norm of each element.

Key-Gen(n)

1. Run setup for the group LWE problem to obtain $\rho : B(n,3) \longrightarrow B(r,3)$
2. Shared key: $SK = \rho$.
3. Using the precomputation, select an element $\tau \in B(r,3)$ of maximal norm.

Symmetric Cryptosystem

We can now proceed with a formal description of the cryptosystem.

Precomputation

Run breadth-first search on the Cayley graph of $B(r, 3)$, recording the norm of each element.

Key-Gen(n)

- Run setup for the group LWE problem to obtain $\varphi : B(n, 3) \longrightarrow B(r, 3)$
- Shared key: $SK = \varphi$
- Using the precomputation, select an element $\tau \in B(r, 3)$ of maximal norm.

We can now proceed with a formal description of the cryptosystem.

Precomputation

Run breadth-first search on the Cayley graph of $B(r, 3)$, recording the norm of each element.

Key-Gen(n)

- Run setup for the group LWE problem to obtain $\varphi : B(n, 3) \longrightarrow B(r, 3)$
- Shared key: SK $\doteq \varphi$
- Using the precomputation, select an element $\tau \in B(r, 3)$ of maximal norm

Symmetric Cryptosystem

We can now proceed with a formal description of the cryptosystem.

Precomputation

Run breadth-first search on the Cayley graph of $B(r,3)$, recording the norm of each element.

Key-Gen(n)

- Run setup for the group LWE problem to obtain $\varphi : B(n,3) \longrightarrow B(r,3)$
- Shared key: SK $\doteq \varphi$
- Using the precomputation, select an element $\tau \in B(r,3)$ of maximal norm

Symmetric Cryptosystem

We can now proceed with a formal description of the cryptosystem.

Precomputation

Run breadth-first search on the Cayley graph of $B(r, 3)$, recording the norm of each element.

Key-Gen(n)

- Run setup for the group LWE problem to obtain $\varphi : B(n, 3) \longrightarrow B(r, 3)$
- Shared key: $\mathrm{SK} \doteq \varphi$
- Using the precomputation, select an element $\tau \in B(r, 3)$ of maximal norm

Symmetric Cryptosystem

We can now proceed with a formal description of the cryptosystem.

Precomputation

Run breadth-first search on the Cayley graph of $B(r, 3)$, recording the norm of each element.

Key-Gen(n)

- Run setup for the group LWE problem to obtain $\varphi : B(n, 3) \longrightarrow B(r, 3)$
- Shared key: SK $\doteq \varphi$
- Using the precomputation, select an element $\tau \in B(r, 3)$ of maximal norm

Symmetric Cryptosystem

Enc(SK, t)

To encrypt a bit t, select $(a, b) \xleftarrow{\$} \mathbf{A}_{\varphi,\Psi_n}$, compute

$$b' \doteq b\tau^t (= \varphi(a)e\tau^t)$$

and output the ciphertext $c \doteq (a, b')$.

Dec(SK, (a, b'))

Compute $e' = \varphi(a)^{-1} \cdot b'$ and output $t = 0$ if and only if $\|e'\| \leq r$.

Symmetric Cryptosystem

Enc(SK, t)

To encrypt a bit t, select $(a, b) \xleftarrow{\$} \mathbf{A}_{\varphi,\Psi_n}$, compute

$$b' \doteq b\tau^t (= \varphi(a) e \tau^t)$$

and output the ciphertext $c \doteq (a, b')$.

Dec(SK, (a, b'))

Compute $e' = \varphi(a)^{-1} \cdot b'$ and output $t = 0$ if and only if $\|e'\| \leq r$.

Correctness

Sketch

For any group G, the norm in the Cayley metric is well-behaved with respect to the group product: for all $a, b \in G$,

$$\left|\|a\| - \|b\|\right| \leq \|ab\| \leq \|a\| + \|b\| .$$

Combining this fact with the Lemma on the diameter, we see that as r grows, correctness is trivial.

(Note: For small r, say $r = 4$, a more careful calculation is required.)

Correctness

Sketch

For any group G, the norm in the Cayley metric is well-behaved with respect to the group product: for all $a, b \in G$,

$$\left| \|a\| - \|b\| \right| \leq \|ab\| \leq \|a\| + \|b\| .$$

Combining this fact with the Lemma on the diameter, we see that as r grows, correctness is trivial.

(Note: For small r, say $r = 4$, a more careful calculation is required.)

Security

Theorem

Under the (decisional) LWE assumption for $B(n, 3)$, the proposed cryptosystem is IND-CPA secure.

Proof Sketch

Given a distinguisher W that differentiates between $\mathbf{E}_0 = \mathrm{Enc}(\mathrm{SK}, 0)$ of encryptions of 0 from $\mathbf{E}_1 = \mathrm{Enc}(\mathrm{SK}, 1)$ of encryptions of 1, construct W' to distinguish $\mathbf{A}_{\varphi,\Psi_n}$ from $\mathbf{U}$ as follows. If given a distribution $\mathbf{R} \in \{\mathbf{A}_{\varphi,\Psi_n}, \mathbf{U}\}$, create two distributions $\mathbf{R}_0 \doteq \mathbf{R}$ and $\mathbf{R}_1 \doteq \mathbf{R} \cdot (1, \tau)$ (*i.e.,* $\mathbf{R}_1$ takes a sample (a, b) from $\mathbf{R}$ and outputs $(a, b\tau)$).

Main point: if $\mathbf{R} = \mathbf{U}$, then $\mathbf{R}_0 = \mathbf{R}_1 = \mathbf{R}$, and if $\mathbf{R} = \mathbf{A}_{\varphi,\Psi_n}$, then $\mathbf{R}_0 = \mathbf{E}_0$ and $\mathbf{R}_1 = \mathbf{E}_1$.

Theorem

Under the (decisional) LWE assumption for $B(n, 3)$, the proposed cryptosystem is IND-CPA secure.

Proof Sketch

Given a distinguisher W that differentiates between $\mathbf{E}_0 = \mathrm{Enc}(\mathrm{SK}, 0)$ of encryptions of 0 from $\mathbf{E}_1 = \mathrm{Enc}(\mathrm{SK}, 1)$ of encryptions of 1, construct W' to distinguish $\mathbf{A}_{\varphi,\Psi_n}$ from $\mathbf{U}$ as follows. If given a distribution $\mathbf{R} \in \{\mathbf{A}_{\varphi,\Psi_n}, \mathbf{U}\}$, create two distributions $\mathbf{R}_0 \doteq \mathbf{R}$ and $\mathbf{R}_1 \doteq \mathbf{R} \cdot (1, \tau)$ (*i.e.,* $\mathbf{R}_1$ takes a sample (a, b) from $\mathbf{R}$ and outputs $(a, b\tau)$).
Main point: if $\mathbf{R} = \mathbf{U}$, then $\mathbf{R}_0 = \mathbf{R}_1 = \mathbf{R}$, and if $\mathbf{R} = \mathbf{A}_{\varphi,\Psi_n}$, then $\mathbf{R}_0 = \mathbf{E}_0$ and $\mathbf{R}_1 = \mathbf{E}_1$.

Work in Progress / Open Questions

- Complexity Reductions (worst case to average case, search to decision)
- Public-key encryption
- Better computational methods for norms in $B(n, 3)$

Work in Progress / Open Questions

- Complexity Reductions (worst case to average case, search to decision)
- Public-key encryption
- Better computational methods for norms in $B(n, 3)$

Work in Progress / Open Questions

- Complexity Reductions (worst case to average case, search to decision)
- Public-key encryption
- Better computational methods for norms in $B(n, 3)$

Fin.

Questions?

Public-Key Encryption?

The techniques of [Reg05] allow parties without any secret information to sample $\mathbf{A}_{\varphi,\Psi}$ (or something close) via subset sums. Doesn't seem to apply in the non-commutative setting:

Observations

- Commutativity allows parties w/o private key to sample instances

$$\sum(\mathbf{s}\cdot\mathbf{a}_i + e_i) = \sum(\mathbf{s}\cdot\mathbf{a}_i) + \sum e_i$$

- In the non-commutative case,

$$\prod(\varphi(a_i)e_i) \neq \prod \varphi(a_i) \prod e_i$$

  and hence small e_i is not sufficient for correctness.

Public-Key Encryption?

The techniques of [Reg05] allow parties without any secret information to sample $\mathbf{A}_{\varphi,\Psi}$ (or something close) via subset sums. Doesn't seem to apply in the non-commutative setting:

Observations

- Commutativity allows parties w/o private key to sample instances

$$\sum(\mathbf{s}\cdot\mathbf{a}_i + e_i) = \sum(\mathbf{s}\cdot\mathbf{a}_i) + \sum e_i$$

- In the non-commutative case,

$$\prod(\varphi(a_i)e_i) \neq \prod\varphi(a_i)\prod e_i$$

and hence small e_i is not sufficient for correctness.

Public-Key Encryption?

The techniques of [Reg05] allow parties without any secret information to sample $\mathbf{A}_{\varphi,\Psi}$ (or something close) via subset sums. Doesn't seem to apply in the non-commutative setting:

Observations

- Commutativity allows parties w/o private key to sample instances

$$\sum(\mathbf{s}\cdot\mathbf{a}_i + e_i) = \sum(\mathbf{s}\cdot\mathbf{a}_i) + \sum e_i$$

- In the non-commutative case,

$$\prod(\varphi(a_i)e_i) \neq \prod \varphi(a_i) \prod e_i$$

and hence small e_i is not sufficient for correctness.

Public-Key Encryption?

The techniques of [Reg05] allow parties without any secret information to sample $\mathbf{A}_{\varphi,\Psi}$ (or something close) via subset sums. Doesn't seem to apply in the non-commutative setting:

Observations

- Commutativity allows parties w/o private key to sample instances

$$\sum(\mathbf{s}\cdot\mathbf{a}_i + e_i) = \sum(\mathbf{s}\cdot\mathbf{a}_i) + \sum e_i$$

- In the non-commutative case,

$$\prod(\varphi(a_i)e_i) \neq \prod \varphi(a_i) \prod e_i$$

and hence small e_i is not sufficient for correctness.

Public-Key Encryption?

Possible Remedies

- Perhaps there is a smarter error distribution Ψ?
- Naïve approach of restricting the support of Ψ to the center of the group is not promising
- More generally, the error terms should not be contained in any proper normal subgroup

Public-Key Encryption?

Possible Remedies

- Perhaps there is a smarter error distribution Ψ?
- Naïve approach of restricting the support of Ψ to the center of the group is not promising
- More generally, the error terms should not be contained in any proper normal subgroup

Public-Key Encryption?

Possible Remedies

- Perhaps there is a smarter error distribution Ψ?
- Naïve approach of restricting the support of Ψ to the center of the group is not promising
- More generally, the error terms should not be contained in any proper normal subgroup

Public-Key Encryption?

Possible Remedies

- Perhaps there is a smarter error distribution Ψ?
- Naïve approach of restricting the support of Ψ to the center of the group is not promising
- More generally, the error terms should not be contained in any proper normal subgroup