Structural Lattice Reduction: Generalized Worst-Case to Average-Case Reductions and Homomorphic Cryptosystems

Nicolas Gama, Malika Izabachene, Phong Nguyen, Xiang Xie

Inpher, Université Paris Saclay, Cea, Inria, CNRS, University of Tokyo, Huawei

May 10, 2016

Section 1

Introduction

In this paper / In this talk

- Generalizing SIS and LWE to arbitrary groups: **duality** aspects.
- Structural reduction: finding short basis in overlattices in poly. time
  - Direct worst to avg.-case reduction for SIS/LWE for arbitrary groups
  - Group-switching: which parameters actually matter for the security of SIS/LWE
  - Self worst-case to average case reducibility of **general** lattice problems
- An abstraction framework that connects lattice based cryptography to classical crypto building blocs
- A fully homomorphic generalization of [GSW13] using our abstraction
  - Link with binary decision diagrams and automata theory.

In this paper / In this talk

- Generalizing SIS and LWE to arbitrary groups: **duality** aspects.
- Structural reduction: finding short basis in overlattices in poly. time
  - Direct worst to avg.-case reduction for SIS/LWE for arbitrary groups
  - Group-switching: which parameters actually matter for the security of SIS/LWE
  - Self worst-case to average case reducibility of **general** lattice problems
- An abstraction framework that connects lattice based cryptography to classical crypto building blocs
- A fully homomorphic generalization of [GSW13] using our abstraction
  - Link with binary decision diagrams and automata theory.

Lattice Problems in Crypto

One actually deals with problems not defined using lattices:

- SIS.
- LWE.

Both are connected to lattice problems
and usually presented with linear algebra:

instead, we adopt a group-theoretical point of view,
and clarify their duality.

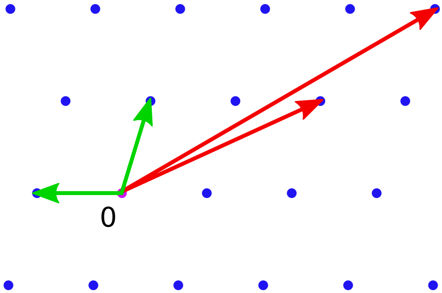
Section 2

Lattice problems, SIS, LWE

Subsection 1

Lattice-based security

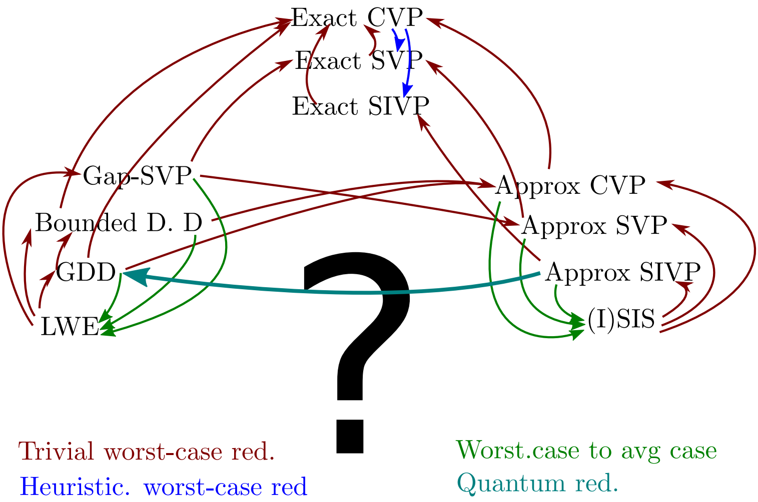
Definition of a lattice



Definition

- Lattice = Discrete subgroup of $\mathbb{R}^m$
- Description: (non-unique) basis.

Lattice problems are a mess



Let's keep it simple...

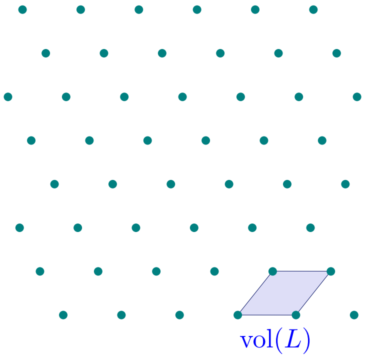
KISS

Keep. It Simple. Stupid.

Finding a lattice point within a ball

- Let L be a lattice and $\mathcal{S}$ a ball, find a lattice point in this ball.
  - L described by a basis
  - $\mathcal{S}$ described by its center and radius.

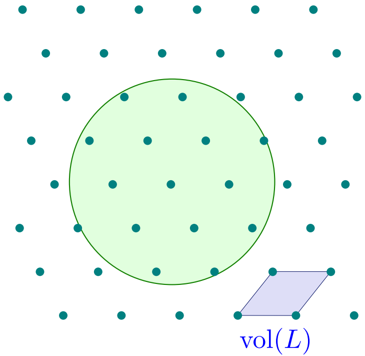
Three categories of problem



Three categories of problem

Approx problem

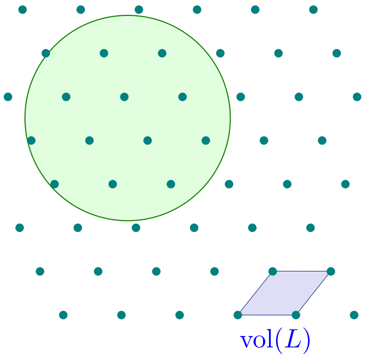
- $\mathrm{vol}(\mathcal{S}) \gg \mathrm{vol}(L)$
  - lots of solutions.



Three categories of problem

Approx problem

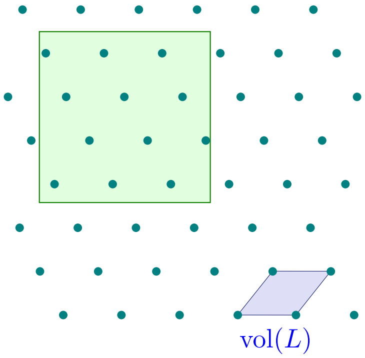
- $\text{vol}(\mathcal{S}) \gg \text{vol}(L)$
  - lots of solutions.



Three categories of problem

Approx problem

- $\mathrm{vol}(\mathcal{S}) \gg \mathrm{vol}(L)$
  - lots of solutions.



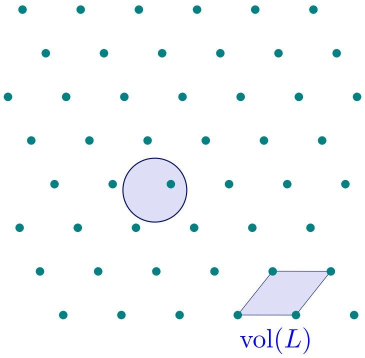
Three categories of problem

Approx problem

- $\mathrm{vol}(\mathcal{S}) \gg \mathrm{vol}(L)$
  - lots of solutions.

Exact problem

- $\mathrm{vol}(\mathcal{S}) \approx \mathrm{vol}(L)$
  - $\approx$single solution



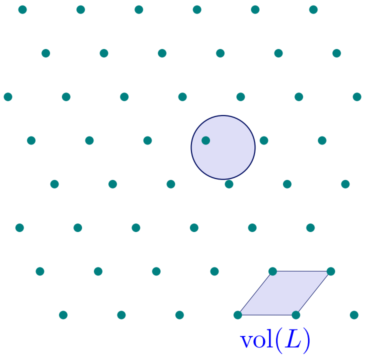
Three categories of problem

Approx problem

- $\mathrm{vol}(\mathcal{S}) \gg \mathrm{vol}(L)$
  - lots of solutions.

Exact problem

- $\mathrm{vol}(\mathcal{S}) \approx \mathrm{vol}(L)$
  - $\approx$single solution



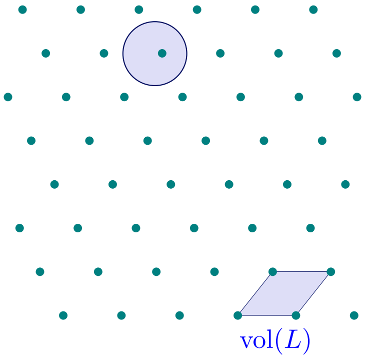
Three categories of problem

Approx problem

- $\mathrm{vol}(\mathcal{S}) \gg \mathrm{vol}(L)$
  - lots of solutions.

Exact problem

- $\mathrm{vol}(\mathcal{S}) \approx \mathrm{vol}(L)$
  - $\approx$single solution



Three categories of problem

Approx problem

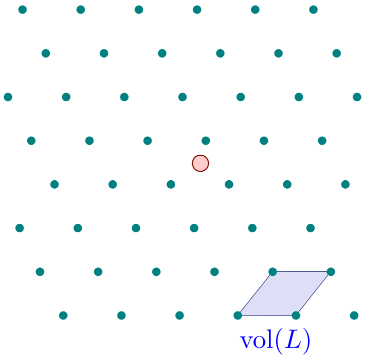
- $\mathrm{vol}(\mathcal{S}) \gg \mathrm{vol}(L)$
  - lots of solutions.

Exact problem

- $\mathrm{vol}(\mathcal{S}) \approx \mathrm{vol}(L)$
  - $\approx$single solution

Unique problem

- $\mathrm{vol}(\mathcal{S}) \ll \mathrm{vol}(L)$
- Random instances have no solution



Three categories of problem

Approx problem

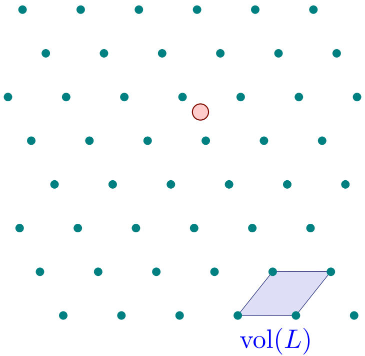
- $\mathrm{vol}(\mathcal{S}) \gg \mathrm{vol}(L)$
  - lots of solutions.

Exact problem

- $\mathrm{vol}(\mathcal{S}) \approx \mathrm{vol}(L)$
  - $\approx$single solution

Unique problem

- $\mathrm{vol}(\mathcal{S}) \ll \mathrm{vol}(L)$
- Random instances have no solution



Three categories of problem

Approx problem

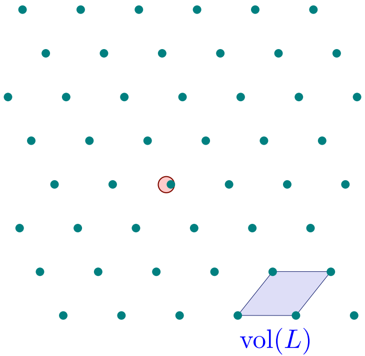
- $\mathrm{vol}(\mathcal{S}) \gg \mathrm{vol}(L)$
  - lots of solutions.

Exact problem

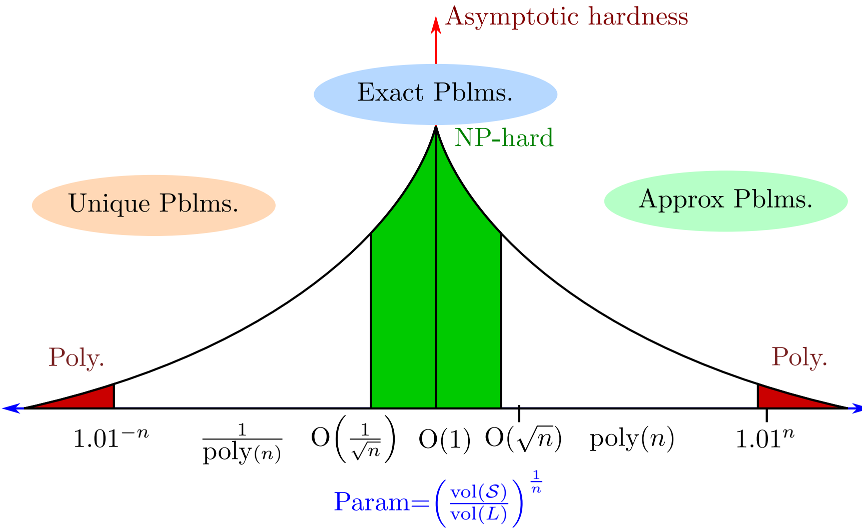
- $\mathrm{vol}(\mathcal{S}) \approx \mathrm{vol}(L)$
  - $\approx$single solution

Unique problem

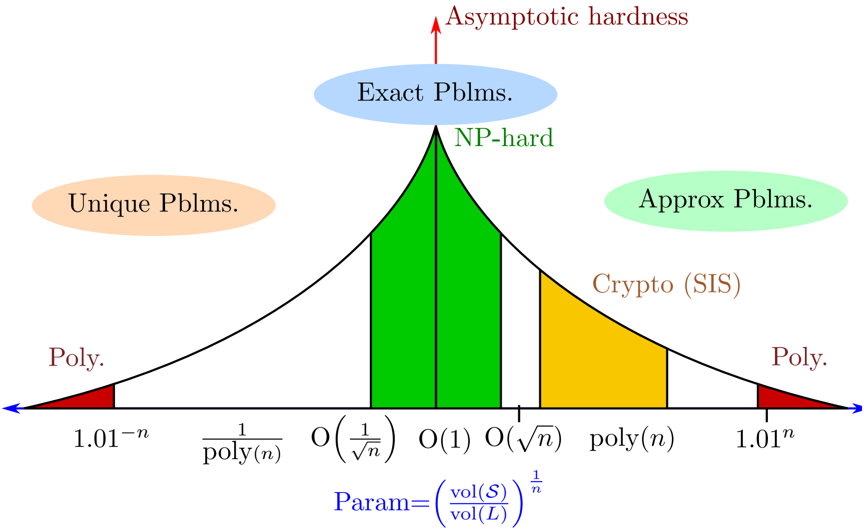
- $\mathrm{vol}(\mathcal{S}) \ll \mathrm{vol}(L)$
- Random instances have no solution
- Only specially crafted instances have a single one



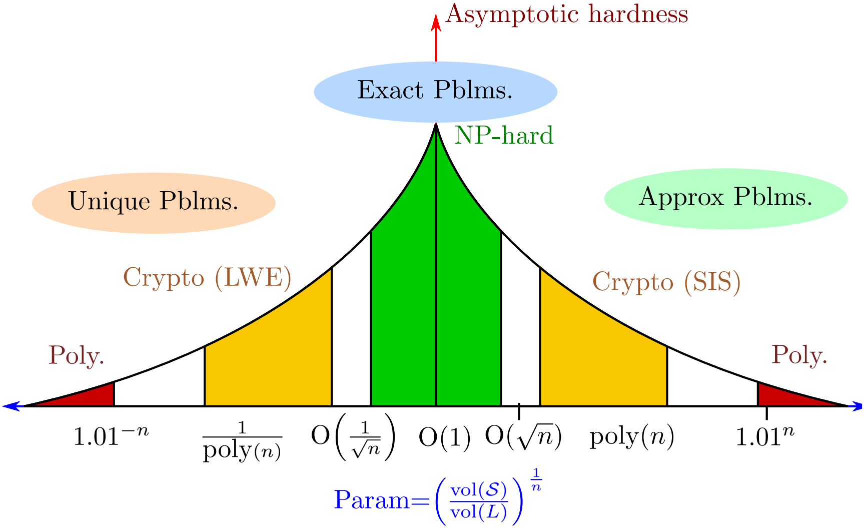
Density vs Proved Asymptotic Hardness



Density vs Proved Asymptotic Hardness



Density vs Proved Asymptotic Hardness



Subsection 2

GSIS

The SIS function over groups

Parameters:

- Pick $g_1, \ldots, g_m$ uniformly at random in an abelian group G

The GSIS function

$$f_{\mathrm{GSIS}}: \quad \begin{array}{rcl} \mathrm{Ball}_\beta(\mathbb{Z}^m) & \rightarrow & G \\ (x_1, \ldots, x_m) & \mapsto & \sum_{i=1}^m x_i g_i \end{array}$$

Properties

- **One way.**
- Inverting f_{GSIS} is the GSIS problem (aka. subset sum, ...)

Link with lattices

Solving GSIS in average is essentially:

finding short vectors in a (uniform) random lattice of

$$L(G) = \{\text{lattices } L \subset \mathbb{Z}^m \text{ s.t. } \mathbb{Z}^m/L \sim G\}$$

Worst-case to Average-case Reduction

[Ajtai96] If one can efficiently solve SIS for $G = (\mathbb{Z}/q\mathbb{Z})^n$ on the average, then one can efficiently find short vectors in every n-dim lattice.

[GINX16] This can be generalized to any finite abelian group G, provided that $\#G$ is sufficiently large $\geq n^{\Omega(\max(n,\mathrm{rank}(G)))}$
Note: $(\mathbb{Z}/2\mathbb{Z})^n$ is not.

Hardness of the GSIS problem

GSIS (Group) hardness depends

- ✔ on the order $\#G$? — ***Yes:*** *harder when* $\#G \nearrow$
- ✔ on β? — ***Yes:*** *harder when* $\beta \searrow$

Hardness of the GSIS problem

GSIS (Group) hardness depends

- ✔ on the order $\#G$? — ***Yes:*** *harder when* $\#G \nearrow$
- ✔ on β? — ***Yes:*** *harder when* $\beta \searrow$

GSIS hardness does NOT depend on

- on m? — ***Should be:*** *harder when* $m \searrow$
  but sometimes, GSIS is intractable $\forall m$
- ✘ on the structure (cycles) of G?
  No! *All structures are equivalent (Structural reduction)*
- ✘ on the choice of the family $(g_1, \ldots, g_m)$?
  No! *Almost all instances are hard (worst-case to avg case red.)*

Subsection 3

GLWE

Duality...

- A character of G is a morphism from G to the torus $\mathbb{T} = (\mathbb{R}/\mathbb{Z}, +)$
- G is isomorphic to its dual group $\hat{G} = \{\text{characters of } G\}$

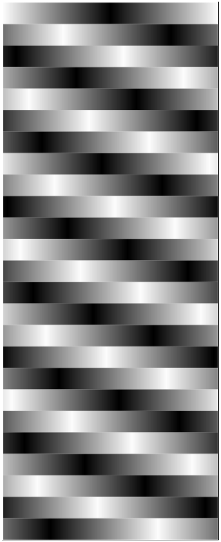
The LWE problem (Regev2005)

Let $(G, +)$ be a finite Abelian group:

- Pick $g_1, \ldots, g_m$ uniformly at random from G.
- Pick a random character $\hat{s}$ in $\hat{G}$.
- **Goal:** recover $\hat{s}$ given $g_1, \ldots, g_m$ and noisy approximations of $\hat{s}(g_1)$, ..., $\hat{s}(g_m)$, where the noise is Gaussian.

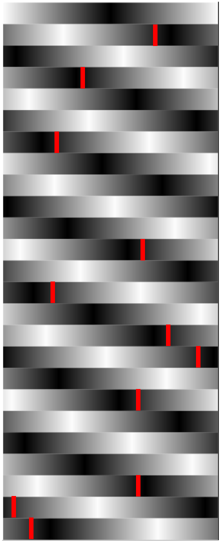
[Regev05] used $G = (\mathbb{Z}/q\mathbb{Z})^n$ like [Ajtai96] for SIS.

Example: cyclic group



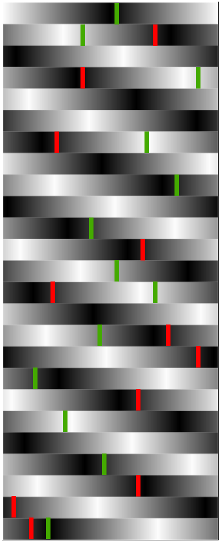
- $g_1, \ldots, g_m = (1, 3, 6, ..., 24)$ rand. in $\mathbb{Z}_{25}$
- secret: $\hat{s}(a) = \frac{2 \cdot a}{25} \mod 1.$

Example: cyclic group



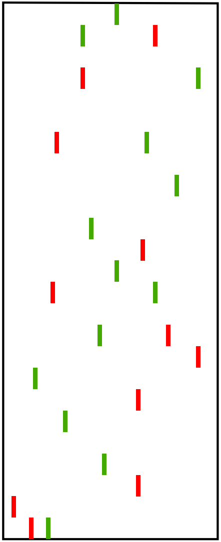
- $g_1, \ldots, g_m = (1, 3, 6, ..., 24)$ rand. in $\mathbb{Z}_{25}$
- secret: $\hat{s}(a) = \frac{2 \cdot a}{25} \mod 1.$
  - GLWE samples $f_{\mathrm{GLWE}}(\hat{s})$

Example: cyclic group



- $g_1, \ldots, g_m = (1, 3, 6, ..., 24)$ rand. in $\mathbb{Z}_{25}$
- secret: $\hat{s}(a) = \frac{2 \cdot a}{25} \mod 1.$
  - GLWE samples $f_{\mathrm{GLWE}}(\hat{s})$
  - Random samples in $\mathbb{T}$

Example: cyclic group



- $g_1, \ldots, g_m = (1, 3, 6, ..., 24)$ rand. in $\mathbb{Z}_{25}$
- secret: $\hat{s}(a) = \frac{2 \cdot a}{25} \mod 1$.
  - GLWE samples $f_{\text{GLWE}}(\hat{s})$
  - Random samples in $\mathbb{T}$

Without the secret ($\hat{s} = \hat{2}$ here)

- Both distributions are very hard to distinguish

The LWE function over groups

Parameters:

- Pick $g_1, \ldots, g_m$ uniformly at random in an abelian group G

The GLWE function

$$\begin{array}{rccl} f_{\mathrm{GLWE}}: & \hat{G} & \to & \mathbb{T}^m \\ & \hat{s} & \mapsto & (\hat{s}(g_i), \ldots, \hat{s}(g_m)) + \text{noise} \end{array}$$

Properties

- **One way.**
- Inverting f_{GLWE} is the GLWE problem

Worst-case to average case reduction

[Regev05]: If one can efficiently solve LWE for $G = (\mathbb{Z}/q\mathbb{Z})^n$ on the average, then one can quantum-efficiently find short vectors in every n-dim lattice.

[GINX16]: This can be generalized to any finite abelian group G, provided that $\#G$ is sufficiently large.

Section 3

Lattice Cryptography

Two Types of Techniques

- Cryptography using trapdoors, i.e. secret short basis of a lattice. Similarities with RSA/Rabin cryptography.
- Cryptography without trapdoors. Similarities with Diffie-Hellmann-based cryptography.

Remember RSA

One-way function

Then $m \to m^e$ is a trapdoor one-way permutation over $(\mathbb{Z}/N\mathbb{Z})^*$.

Trapdoor

- $d = e^{-1} \mod \phi(N)$ is a trapdoor.
- Very expensive to compute from (N, e), but once we have it, inversion $c \to c^d$ is easy
- *(in general we build the trapdoor first!)*

Trapdoor for Lattices

One way functions

- f_{GSIS}: short $(x_1 \ldots, x_m) \rightarrow \sum x_i g_i$
- f_{GLWE}: character $\hat{s} \rightarrow (\hat{s}(g_1), \ldots, \hat{s}(g_m)) + \text{noise}$

Trapdoor

- But if we get any short basis of the SIS lattice $(g_1, \ldots, g_m)^\perp$, both become easy to invert.
  see: [GGH97], [Micc01], [NTRU96], [GPV08], [MP12], [CGGI16]...
- *(Again, one would build the trapdoor first!)*

Trapdoor-less Diffie Hellmann

Alice

Bob

Trapdoor-less Diffie Hellmann

Alice

Bob

$G = \langle g \rangle$
of order q

$g^a \in G$

$a \in \mathbb{Z}_q$

Trapdoor-less Diffie Hellmann

Alice

Bob

$G = \langle g \rangle$
of order q

$a \in \mathbb{Z}_q$

$g^a \in G$

$b \in \mathbb{Z}_q$

$g^b \in G$

Trapdoor-less Diffie Hellmann

Alice

Bob

$G = \langle g \rangle$
of order q

$g^a \in G$

$a \in \mathbb{Z}_q$

$b \in \mathbb{Z}_q$

$g^b \in G$

$g^{ab} \in G$

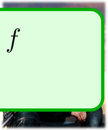
???!!

$g^{ab} \in G$

Trapdoor-less Diffie Hellmann

Alice

Bob



A one-way function f

$f(a) = g^a$

$g^a \in G$

$a \in \mathbb{Z}_q$

$b \in \mathbb{Z}_q$

A pairing e

$e(a,b) = g^{ab}$

that can be computed even if

a or b is hidden by f

$g^{ab} \in G$

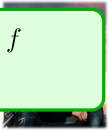
???!!

$g^{ab} \in G$

Trapdoor-less Diffie Hellmann

Alice

Bob



A one-way function f

$f(a) = g^a$

$g^a \in G$

$a \in \mathbb{Z}_q$

$b \in \mathbb{Z}_q$

A pairing e

$e(a, b) = g^{ab}$

that can be computed even if

a or b is hidden by f

$g^{ab} \in G$

???!!

$g^{ab} \in G$

- This key exchange is the core of El-Gamal PK encryption

Abstracting DH

- let $e : e(a,b) = g^{ab}$ this map is a pairing, it is bilinear from $\mathbb{Z}_q \times \mathbb{Z}_q \to G$.
- let $f : \ a \mapsto g^a$ be the DL one-way function

Computability

$e(a,b)$ can be computed using $(f(a), b)$ or $(a, f(b))$
even if a or b is hidden by f

Security

hard to distinguish $(f(a), f(b), e(a,b))$ from $(f(a), f(b), \mathrm{random})$

DH with lattices

- What would be the pairing?
- What would be the one-way function to hide inputs?

DH with lattices

The Pairing (for $g_1, \dots, g_m \leftarrow_{\$} G$)

$$\begin{array}{rccc} \xi: & \hat{G} \times \mathbb{Z}^m & \to & \mathbb{T} \\ & (\hat{s}, \mathbf{x}) & \mapsto & \hat{s}\left(\sum_{i=1}^{m} x_i g_i\right) \end{array}$$

errors/approximations are ok if x_i are small.

Computability

- Let $y = f_{\mathrm{SIS}}(x_1, \dots, x_m) = \sum x_i g_i \in G$
  Then: $\xi(\hat{s}, \mathbf{x}) = \hat{s}(y) \in \mathbb{T}$
- Let $\mathbf{b} = f_{\mathrm{LWE}}(\hat{s}) \approx (\hat{s}(g_1), \dots, \hat{s}(g_m)) \in \mathbb{T}^m$
  Then: $\xi(\hat{s}, \mathbf{x}) \approx \langle \mathbf{x}, \mathbf{b} \rangle \in \mathbb{T}$

Noisy key exchange from lattices

Alice

Bob

Noisy key exchange from lattices

Alice

$\mathbf{x} \in \mathbb{Z}^m$

$g_1, \ldots, g_m \in G$

$y = f_{\mathrm{GSIS}}(\mathbf{x})$

Bob

Noisy key exchange from lattices

Alice

Bob

$g_1, \ldots, g_m \in G$

$\mathbf{x} \in \mathbb{Z}^m$

$y = f_{\text{GSIS}}(\mathbf{x})$

$\hat{s} \in \hat{G}$

$\mathbf{b} = f_{\text{GLWE}}(\hat{s})$

Noisy key exchange from lattices

Alice

Bob

$g_1, \ldots, g_m \in G$

$\mathbf{x} \in \mathbb{Z}^m$

$y = f_{\mathrm{GSIS}}(\mathbf{x})$

$\hat{s} \in \hat{G}$

$\mathbf{b} = f_{\mathrm{GLWE}}(\hat{s})$

$\xi(\hat{s}, \mathbf{x}) \approx \langle \mathbf{x}, \mathbf{b} \rangle$

???!!

$\xi(\hat{s}, \mathbf{x}) = \hat{s}(y)$

(noise can be eliminated by rounding)

El-Gamal encryption from lattices

- This key exchanges gives rise to two El-Gamal PK encryptions schemes, because the lattice pairing is not symmetric
- These El-Gamal-like schemes are IND-CPA secure under the hardness of SIS/LWE (post-quantum?)
- Similarly, many LWE/SIS schemes can be viewed as analogues of the RSA/DL world: [GPV08] is a lattice analogue of Rabin's signature, etc.

Section 4

Fully Homomorphic Encryption

Among the various homomorphic schemes:

- [Gentry09], [BGV], [SHE], [YaSHE], [GSW], ...

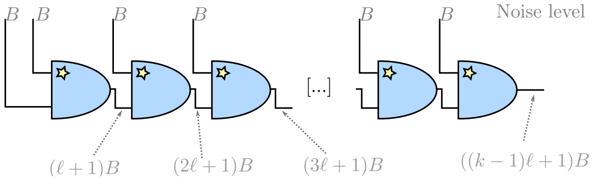
We focused on one particular line

- [GSW13] - [AP14] - [DM15]

What can it do

GSW can do

- (Like all others) additions and linear combinations
- Based on LWE (see f_{LWE} + trapdoors)
- [AP14]: Conjunctions with sublinear noise overhead!
- [DM15]: (external) Bootstrapping in less than 1 second!



What we add

[GINX16]: evaluate (reduced) binary decision diagrams and deterministic automata with sublinear noise overhead, (almost) independently of the depth!!

[GINX16]: universal composition of boolean functions (with exponential noise overhead in the number of compositions)

[GINX16]: internal (but slow) bootstrapping

Why it matters

it really matters a lot!

- We implement our everyday's life problems as finite state machine algorithms:
    - see elementary school arithmetic: equality, an order check, an addition, a multiplication...
    - but also non arithmetic: full-text search, password check, etc...!
- Having the automata or binary decision diagram logic is way more general than polynomial arithmetic

The main drawback

If it is so "perfect", why don't we use it in practice?

- The gate complexity is polynomial in the security parameter $\approx O(\lambda^2)$ per bit, with very limited batching capabilities
- In general (except for [DM15] bootstrapping), performance of the whole GSW line lags **very far behind** other candidates (BGV, YaSHE, ...)

The main drawback

If it is so "perfect", why don't we use it in practice?

- The gate complexity is polynomial in the security parameter $\approx O(\lambda^2)$ per bit, with very limited batching capabilities
- In general (except for [DM15] bootstrapping), performance of the whole GSW line lags **very far behind** other candidates (BGV, YaSHE, ...)

So... performance = big open problem?

- Don't despair, stay tuned, work is in progress!

Summary/Conclusion

1. Direct worst-case to average reductions for any group structures, via a new simple tool call “structural-reduction”
2. A framework to abstract Lattice cryptography and to link it with traditional crypto constructions.
    1. allows to build new post-quantum cryptosystems
    2. also to transfer security properties from traditional systems to lattice-based ones!
3. A new framework for LWE-based homomorphic encryption:
    - based on classical automata and binary decision diagram theories
    - and universal composition of boolean functions