Random Self-reducibility of Ideal-SVP via Arakelov Random Walks

Koen de Boer[1] and Léo Ducas[1]
and **Alice Pellet-Mary**[2] and Benjamin Wesolowski[2]

[1] CWI, Amsterdam [2] CNRS and Université de Bordeaux

Séminaire de théorie des nombres de Toulouse

https://eprint.iacr.org/2020/297.pdf

Context

For public key cryptography, we need hard algorithmic problems

Context

For public key cryptography, we need hard algorithmic problems

What does hard mean?

$\rightsquigarrow$ we don't know any polynomial time algorithm that solves the problem

Context

For public key cryptography, we need hard algorithmic problems

What does hard mean?

$\rightsquigarrow$ we don't know any polynomial time algorithm that solves the problem

Examples:

- Factoring
- Discrete logarithm

Foundation of public key cryptography

Cryptographic primitives (public key)

public key encryption | signature | homomorphic encryption | …

error correcting codes | lattices | isogenies

factoring | discrete logarithm | …

Supposedly intractable algorithmic problems

Foundation of public key cryptography

Cryptographic primitives (public key)

public key encryption | signature | homomorphic encryption | . . .

error correcting codes | lattices | isogenies

~~factoring~~ | ~~discrete logarithm~~ | . . .

Supposedly intractable algorithmic problems in a quantum world

Foundation of public key cryptography

Cryptographic primitives (public key)

public key encryption | signature | homomorphic encryption | . . .

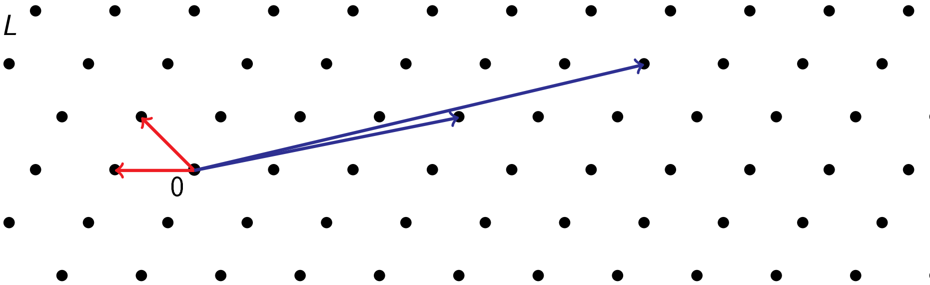
error correcting codes | lattices | isogenies

~~factoring~~ | ~~discrete logarithm~~ | . . .

Supposedly intractable algorithmic problems in a quantum world

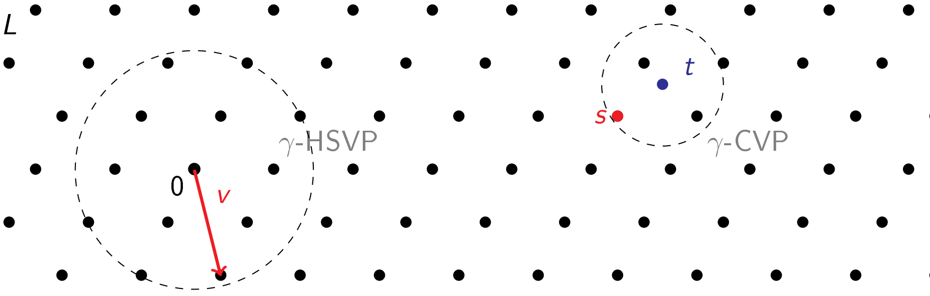
Lattices

Lattices



- $L = \{Bx \mid x \in \mathbb{Z}^n\}$ is a lattice
- $B \in \mathrm{GL}_n(\mathbb{R})$ is a basis
- n is the dimension of L
- $|\det(B)| =: \mathrm{Vol}(L)$ is the volume of L (does not depend on the basis B)
  - in this talk $\mathrm{Vol}(L) = 1$ always

Algorithmic problems



γ-HSVP

(Hermite Shortest Vector Problem)

Find $v \in L$ such that $\|v\|_2 \leq \gamma$

γ-CVP

(Closest Vector Problem)

Given $t \in \mathbb{R}^n$, find $s \in L$ such that $\|t - s\|_2 \leq \gamma$

(**input:** a basis of L)

Hardness of HSVP and CVP

γ-HSVP and γ-CVP are hard to solve

- if the input is a bad basis of L
- if $\gamma = \mathrm{poly}(n)$
- in the worst case
  - we don't have a polynomial time algorithm that works for all lattices

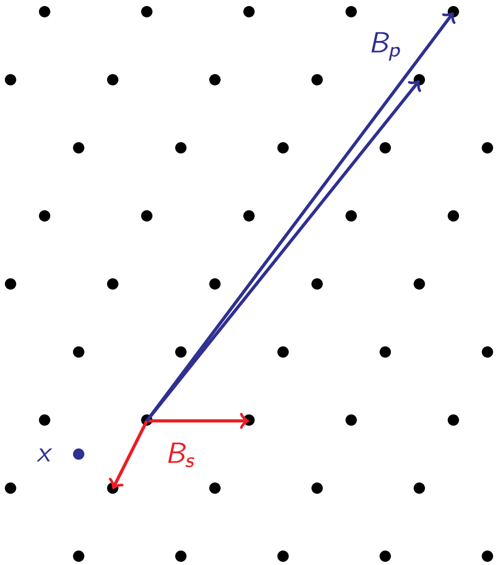
Hardness of HSVP and CVP

γ-HSVP and γ-CVP are hard to solve

- if the input is a bad basis of L
- if $\gamma = \mathrm{poly}(n)$
- in the worst case
  - we don't have a polynomial time algorithm that works for all lattices

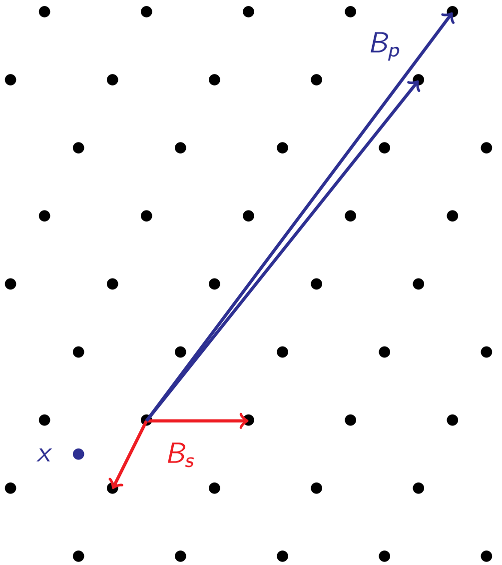
Remark: if we have a good basis of L, then they become easy

Public key encryption from lattices



$\mathrm{pk} = (B_p, x)$
$\mathrm{sk} = B_s$

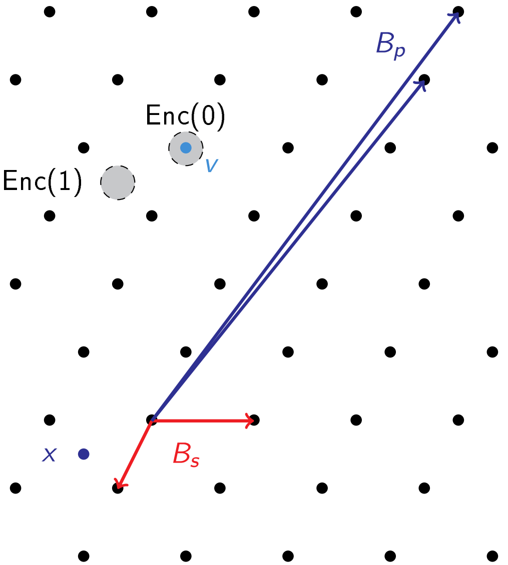
Public key encryption from lattices



$\mathrm{pk} = (B_p, x)$
$\mathrm{sk} = B_s$

message: $m \in \{0, 1\}$

Public key encryption from lattices



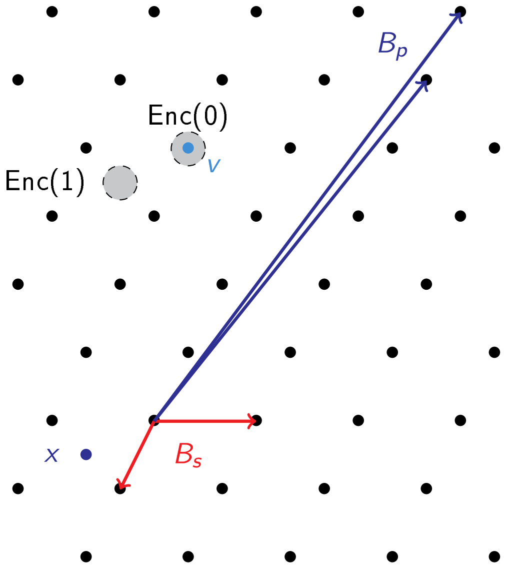
$\mathrm{pk} = (B_p, x)$
$\mathrm{sk} = B_s$

message: $m \in \{0, 1\}$

Encryption(m, pk):

- sample random $v \in L$
- sample small $e \in \mathbb{R}^n$
- return $c = v + e + m \cdot x$

Public key encryption from lattices



$\mathrm{pk} = (B_p, x)$
$\mathrm{sk} = B_s$

message: $m \in \{0, 1\}$

Encryption(m, pk):

- sample random $v \in L$
- sample small $e \in \mathbb{R}^n$
- return $c = v + e + m \cdot x$

Decryption(c, sk):

- find $w \in L$ closest to c
- if c is very close to w, return $m = 0$
- otherwise return $m = 1$

Summary (so far)

- we need hard algorithmic problems for cryptography
- γ-HSVP is such a hard problem

From now on, we focus on γ-HSVP

γ-HSVP: given a bad basis of a lattice L (with $\mathrm{vol}(L) = 1$), find $v \in L$ such that $\|v\|_2 \leq \gamma$

Ideal lattices

Why?

Motivation

Schemes using lattices are usually not efficient
(storage: n^2, matrix-vector mult: n^2)
$\Rightarrow$ improve efficiency using ideal lattices

Why?

Motivation

Schemes using lattices are usually not efficient
(storage: n^2, matrix-vector mult: n^2)
$\Rightarrow$ improve efficiency using ideal lattices

$$M_a = \begin{pmatrix} a_1 & -a_n & \cdots & -a_2 \\ a_2 & a_1 & \cdots & -a_3 \\ \vdots & \ddots & \ddots & \vdots \\ a_n & a_{n-1} & \dots & a_1 \end{pmatrix}$$

basis of a special case of
ideal lattice

Some definitions

Notation

$K = \mathbb{Q}[X]/(X^n+1)$, with $n = 2^k$ (or any number field)

$O_K = \mathbb{Z}[X]/(X^n+1)$

$K_\mathbb{R} = K \otimes_\mathbb{Q} \mathbb{R} = \mathbb{R}[X]/(X^n+1)$

Some definitions

Notation

$K = \mathbb{Q}[X]/(X^n + 1)$, with $n = 2^k$ (or any number field)

$O_K = \mathbb{Z}[X]/(X^n + 1)$

$K_{\mathbb{R}} = K \otimes_{\mathbb{Q}} \mathbb{R} = \mathbb{R}[X]/(X^n + 1)$

- integral ideal: $\mathfrak{a} \subseteq O_K$
- oriented replete ideal: $I := \alpha \cdot \mathfrak{a} \subset K_{\mathbb{R}}$, with $\alpha \in K_{\mathbb{R}}$ and $\mathfrak{a} \subseteq O_K$
  (e.g., $I = \sqrt{2} \cdot \langle 3 \rangle = \{\sqrt{2} \cdot 3 \cdot x \mid x \in \mathbb{Z}\} \subset \mathbb{R}$)

Some definitions

Notation

$K = \mathbb{Q}[X]/(X^n + 1)$, with $n = 2^k$ (or any number field)

$O_K = \mathbb{Z}[X]/(X^n + 1)$

$K_\mathbb{R} = K \otimes_\mathbb{Q} \mathbb{R} = \mathbb{R}[X]/(X^n + 1)$

- integral ideal: $\mathfrak{a} \subseteq O_K$
- oriented replete ideal: $I := \alpha \cdot \mathfrak{a} \subset K_\mathbb{R}$, with $\alpha \in K_\mathbb{R}$ and $\mathfrak{a} \subseteq O_K$
  (e.g., $I = \sqrt{2} \cdot \langle 3 \rangle = \{\sqrt{2} \cdot 3 \cdot x \,|\, x \in \mathbb{Z}\} \subset \mathbb{R}$)

From now on:

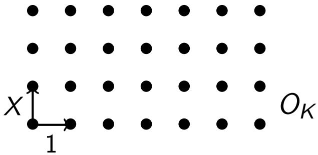
- ideal := oriented replete ideal
- I is an ideal
- $\mathcal{N}(I) = 1$

Why is I a lattice?

O_K is a lattice

$$\begin{aligned} \sigma : O_K = \mathbb{Z}[X]/(X^n+1) &\to \mathbb{C}^n \\ r(X) &\mapsto (r(\alpha_1), r(\alpha_2), \dots, r(\alpha_n)), \end{aligned}$$

where $\alpha_1, \dots, \alpha_n$ are the roots of $X^n + 1$ in $\mathbb{C}$



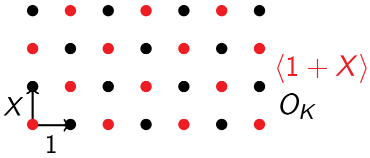
Why is I a lattice?

O_K is a lattice

$$\sigma : O_K = \mathbb{Z}[X]/(X^n + 1) \rightarrow \mathbb{C}^n$$
$$r(X) \mapsto (r(\alpha_1), r(\alpha_2), \ldots, r(\alpha_n)),$$

where $\alpha_1, \ldots, \alpha_n$ are the roots of $X^n + 1$ in $\mathbb{C}$

$$\begin{cases} \sigma(I) \subseteq \sigma(O_K) \simeq \mathbb{Z}^n \\ \text{stable by '+' and '−'} \end{cases} \Rightarrow \text{ideal lattice}$$



γ-ideal-HSVP

γ-ideal-HSVP = γ-HSVP restricted to ideal lattices

γ-ideal-HSVP: given a basis of an ideal lattice $\sigma(I)$ (with $\mathcal{N}(I) = 1$) , find $x \in I$ such that $\|\sigma(x)\|_2 \leq \gamma$.

γ-ideal-HSVP

γ-ideal-HSVP = γ-HSVP restricted to ideal lattices

γ-ideal-HSVP: given a basis of an ideal lattice $\sigma(I)$ (with $\mathcal{N}(I) = 1$) , find $x \in I$ such that $\|\sigma(x)\|_2 \leq \gamma$.

This is still a hard problem

- if the input basis of $\sigma(I)$ is bad
- if $\gamma = \mathrm{poly}(d)$
- in the worst case

  (no poly time algorithm that works for all ideal lattices)

Summary (so far)

- we need hard algorithmic problems for cryptography
- γ-HSVP is a hard problem
- γ-HSVP restricted to ideal lattices is still a hard problem

From now on, we focus on γ-ideal-HSVP

γ-ideal-HSVP: given a bad basis of an ideal lattice $\sigma(I)$ (with $\mathcal{N}(I) = 1$) , find $x \in I$ such that $\|\sigma(x)\|_2 \leq \gamma$.

Average-case hardness

Worst-case hardness

γ-ideal-HSVP is hard in the worst case:

- we don't have a polynomial time algorithm that works for all ideals
- but maybe most of the ideals are easy

Worst-case hardness

γ-ideal-HSVP is hard in the worst case:

- we don't have a polynomial time algorithm that works for all ideals
- but maybe most of the ideals are easy

How do we generate ideals I for which γ-ideal-HSVP is hard?

(this is needed for crypto)

Our result

Theorem [BDPW20]

There is a distribution D over ideal lattices such that

solving γ-ideal-HSVP in I with non-negligible probability when $I \leftarrow D$
$\Rightarrow$ solving γ'-ideal-HSVP in all ideals I

with $\gamma' = \sqrt{d} \cdot \gamma$

γ-ideal-HSVP is hard on average.

Our result

Theorem [BDPW20]

There is a distribution D over ideal lattices such that

solving γ-ideal-HSVP in I with non-negligible probability when $I \leftarrow D$
$\Rightarrow$ solving γ'-ideal-HSVP in all ideals I

with $\gamma' = \sqrt{d} \cdot \gamma$

γ-ideal-HSVP is hard on average.

Remark. D is efficiently samplable.

Our result

Theorem [BDPW20]

There is a distribution D over ideal lattices such that

solving γ-ideal-HSVP in I with non-negligible probability when $I \leftarrow D$
$\Rightarrow$ solving γ'-ideal-HSVP in all ideals I

with $\gamma' = \sqrt{d} \cdot \gamma$

γ-ideal-HSVP is hard on average.

Remark. D is efficiently samplable.

We can sample hard ideal lattices for crypto

(very small probability that the sampled ideal is an easy one)

Techniques of the proof

Conclusion

Conclusion

Cryptography needs algorithmic problems that are hard on average

[Gen10] Gentry. Toward basing fully homomorphic encryption on worst-case hardness. Crypto

Conclusion

Cryptography needs algorithmic problems that are hard on average

- ideal-HSVP is believed to be hard in the worst case
- we show that if ideal-HSVP is hard in the worst-case, then it is also hard on average.
  - can be used for crypto

[Gen10] Gentry. Toward basing fully homomorphic encryption on worst-case hardness. Crypto

Conclusion

Cryptography needs algorithmic problems that are hard on average

- ideal-HSVP is believed to be hard in the worst case
- we show that if ideal-HSVP is hard in the worst-case, then it is also hard on average.
  - can be used for crypto
- a worst-case to average-case reduction was already proven in [Gen10]
  - requires a quantum computer
  - worse loss $\gamma \to \gamma'$
  - different distribution D and different proof

[Gen10] Gentry. Toward basing fully homomorphic encryption on worst-case hardness. Crypto

Conclusion

Cryptography needs algorithmic problems that are hard on average

- ideal-HSVP is believed to be hard in the worst case
- we show that if ideal-HSVP is hard in the worst-case, then it is also hard on average.
  - can be used for crypto
- a worst-case to average-case reduction was already proven in [Gen10]
  - requires a quantum computer
  - worse loss $\gamma \to \gamma'$
  - different distribution D and different proof

Thank you

[Gen10] Gentry. Toward basing fully homomorphic encryption on worst-case hardness. Crypto