# INTRODUCTION TO (LATTICE-BASED) CRYPTOGRAPHY

**Adeline Roux-Langlois**

Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, Caen, FRANCE

GREYC
Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

cnrs

# Cryptography

Let's start with a simple example: you want to send a message to someone.

Two possibilities:

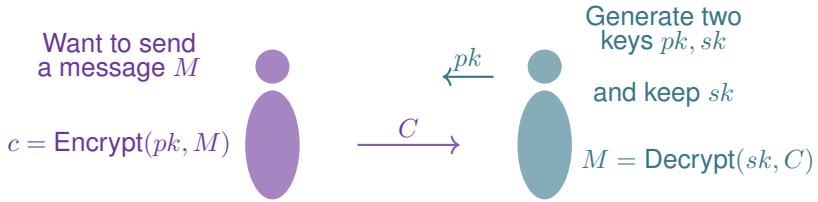
- Either you share a secret key (AES...),
- Either you don't $\Rightarrow$ public key cryptography.

# Cryptography

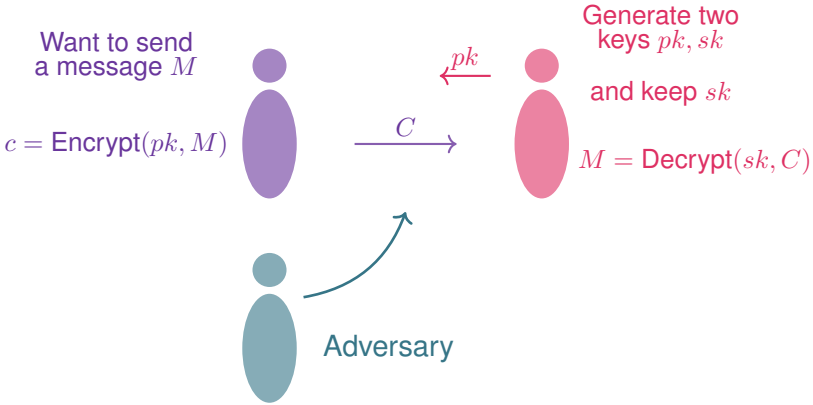
Let's start with a simple example: you want to send a message to someone.

Two possibilities:

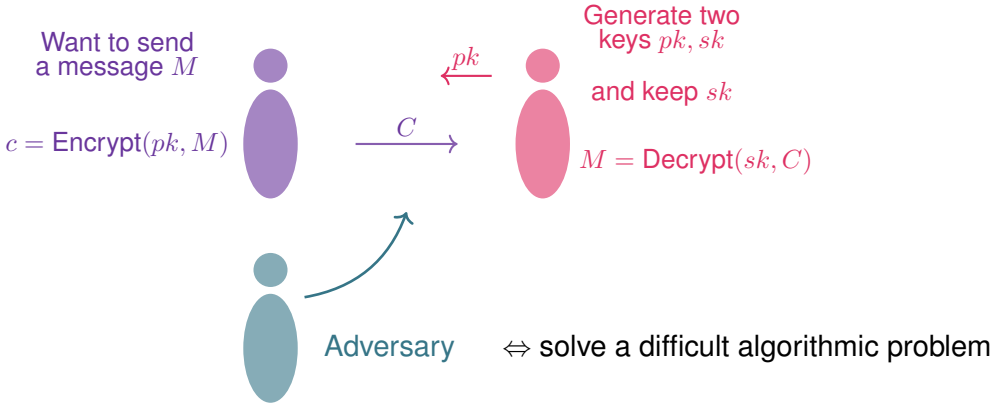
- Either you share a secret key (AES...),
- Either you don't $\Rightarrow$ public key cryptography.



# Public key cryptography

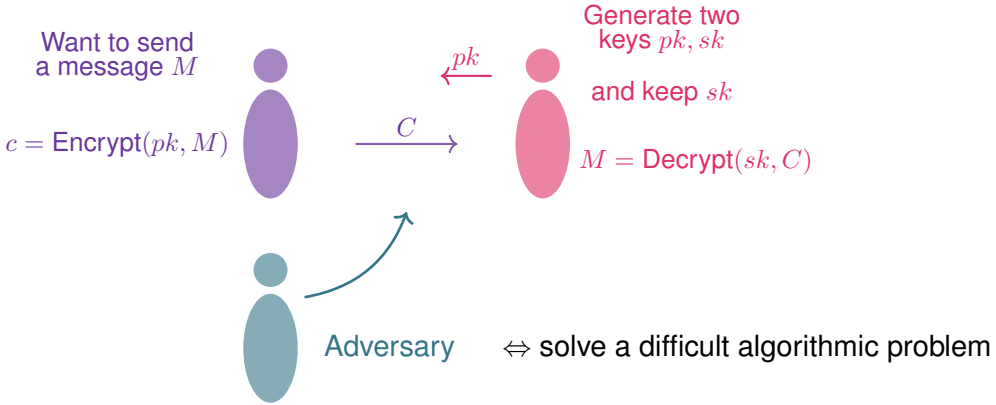


# Public key cryptography



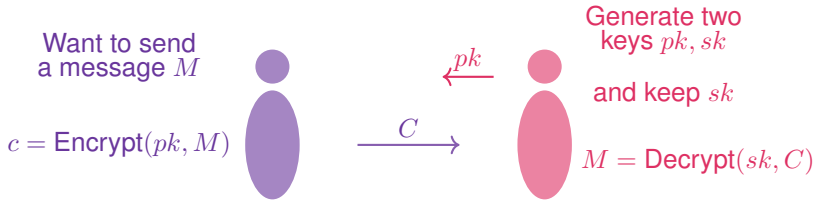

$\Leftrightarrow$ solve a difficult algorithmic problem

# Public key cryptography




- Examples: factorisation (RSA), discrete log (El Gamal) ...
- What does it mean to be difficult? **Solving those problems needs an exponential complexity on a classical computer.**

# Public key encryption




**Definition**

A public key encryption scheme is defined by three algorithms **(KeyGen, Enc, Dec)** such that:

- **KeyGen** takes as input the security parameter $\lambda$ and outputs the keys $(pk, sk)$,
- **Enc** takes as input $pk$ and a message $m$ and outputs $c = Enc(pk, m)$,
- **Dec** takes as input $sk$ and a cipher $c$ and outputs $m = Dec(sk, c)$,

such that $Dec(sk, (Enc(pk, m)) = m$.

# Public key encryption

**Definition**
A public key encryption scheme is defined by three algorithms **(KeyGen, Enc, Dec)** such that:

- **KeyGen** takes as input the security parameter $\lambda$ and outputs the keys $(pk, sk)$,
- **Enc** takes as input the public key $pk$ and a message $m$ and outputs $c = Enc(pk, m)$,
- **Dec** takes as input the secret key $sk$ and a cipher $c$ and outputs $m = Dec(sk, c)$,

such that $Dec(sk, (Enc(pk, m)) = m$.

Two important properties: correctness and security.

# Public key encryption

How do we define security of a public key encryption?

# Public key encryption

How do we define security of a public key encryption?

We use the notion of indistinguishability of the ciphertexts.

# IND-CPA security

To define the security, we use a game between a challenger and an adversary.
We define the following experiment:

| $\mathcal{C}$hallenger | | $\mathcal{A}$dversary |
|---|---|---|
| $b \leftarrow U(\{0,1\})$<br>Generate $(pk, sk)$ | $\xrightarrow{pk}$ | |
| | $\xleftarrow{M_0, M_1}$ | Choose $M_0, M_1$ |
| | $\xrightarrow{c \leftarrow \mathsf{Enc}(pk, M_b)}$ | Output $b'$<br>$\mathcal{A}$ wins if $b = b'$ |

$$Adv^{CPA}(\mathcal{A}) = |\Pr[\mathcal{A} \text{ wins}] - 1/2|$$

# Today: an introduction to (lattice-based) cryptography

1. El Gamal encryption scheme
   - Background
   - Discrete logarithm problem
   - El Gamal scheme and its security

2. Regev's encryption scheme
   - Lattices and hard problem on lattices
   - Learning With Errors problem
   - How to encrypt using LWE?
   - Practical scheme using Module-LWE

# Security hypothesis

To build a public key encryption scheme, we need:

- a "difficult to inverse" problem: allows to build a public key using a secret key but not to come back,
- a efficient key generation.

Example of RSA encryption scheme:

- the secret key is $(p, q)$ two distinct primes
  $\Rightarrow$ we have to be able to generate large prime efficiently
- the public key is $N = pq$
  $\Rightarrow$ the "factorization" problem must be difficult to solve.

# Difficult problem vs efficient algorihtm

Computational security: different from a perfect security,
schemes can be attacked
but it must be difficult (too slow in practice).

Let $n$ be the security parameter:

- Efficient algorithm = polynomial in $n$ ($n^c$ for constant $c$),
- Difficult problem = no algorithm in polynomial time, best know algorithm has complexity exponential in $n$.

Order of magnitude:

- Today, a difficult problem $\Rightarrow$ complexity $2^{80}$ or $2^{128}$
  - A 3.4GHz processor executes $3.4 \times 10^9$ cycles per second
  - $2^{60}$ cycles requires $340 \times 10^6$ secondes (around 11 years),
  - $2^{80}$ is $2^{20}$ (around 1 million) times $2^{60}$ ...

# Background

$$\mathbb{Z}_N = \{0, 1, 2, \ldots, N-1\}$$

- Operations of addition, substraction and multiplication,
- $(\mathbb{Z}_N, +, \times)$ is a ring.

- Inverse: if $a$ is prime with $N$, the multiplicative inverse of $a \bmod N$ is $b$ such that $ab = 1 \bmod N$.
  $\rightarrow$ To find the inverse, use Extended Euclidean algorithm.
  $\rightarrow$ Allows to find $(u, v)$ such that $au + Nv = 1 \bmod N$.

- Particular case: $p$ prime $\rightarrow \mathbb{Z}_p = \{1, 2, \ldots, p-1\}$ has only inversible elements,
  $\rightarrow (\mathbb{Z}_p, +, \times)$ is a field.

$$\mathbb{Z}_N^* = \{a \in \{1, \ldots, N-1\} | \gcd(a, N) = 1\},$$
it is the set of integers of $\mathbb{Z}_N$ invertible modulo $N$.

- Euler's totient function: $\varphi(N) = Card(\mathbb{Z}_N^*)$
  - If $N$ prime: $\varphi(N) = N - 1$,
  - If $N = \prod_i p_i^{e_i}$ with $p_i$ prime and $e_i > 1$, then $\varphi(N) = \prod_i p_i^{e_i - 1}(p_i - 1)$.

**Euler's Theorem**

For $N > 1$ and $x \in \mathbb{Z}_N^*$, we have that $x^{\varphi(N)} = 1 \bmod N$.

Consequence for $x \in \mathbb{Z}_N^*$: $x^a \bmod N = x^{a \bmod \varphi(N)} \bmod N$.

# Discrete logarithm problem

- If $p$ is prime then $\mathbb{Z}_p^*$ is a cyclic group of order $p-1$.
  It means it has $\varphi(p-1)$ generators $g$ such that $\mathbb{Z}_p^* = (1, g, g^2, \cdots, g^{p-2})$.

**Discrete Logarithm Problem (DLP):**
Let $G$ be a finite group ($\mathbb{Z}_p^*$ as example), $q$ its order, $g$ a generator, and $g^a$ where $a$ is uniformly sampled in $\mathbb{Z}_q$, find $a$.

*This problem can be difficult to solve or easy, it depends on the group!*

- Historical choice: $\mathbb{Z}_p^*$ with $p$ prime,
- Bad choice possible,
- Good choice: quadratic residues subgroup of $\mathbb{Z}_p^*$,
- Today: elliptic curves.

# DLP and its variants

**Discrete Logarithm Problem (DLP):**
Let $G$ be a finite group, $q$ its order, $g$ a generator, and $g^a$ where $a$ is uniformly sampled in $\mathbb{Z}_q$, find $a$.

**Computational Diffie-Hellman problem (CDH):**
Let $G$ be a finite group, $q$ its order, $g$ a generator, and $g^a$, $g^b$ where $a$ and $b$ are uniformly sampled in $\mathbb{Z}_q$, compute $g^{ab}$.

**Decisional Diffie-Hellman problem (DDH):**
Given $g$ a generator, distinguish between the distribution $(g^a, g^b, g^{ab})$ and$(g^a, g^b, g^c)$ where $a$, $b$ and $c$ are uniformly sampled in $\mathbb{Z}_q$.

# DLP and its variants

**Discrete Logarithm Problem (DLP):**
Let $G$ be a finite group, $q$ its order, $g$ a generator, and $g^a$ where $a$ is uniformly sampled in $\mathbb{Z}_q$, find $a$.

**Computational Diffie-Hellman problem (CDH):**
Let $G$ be a finite group, $q$ its order, $g$ a generator, and $g^a$, $g^b$ where $a$ and $b$ are uniformly sampled in $\mathbb{Z}_q$, compute $g^{ab}$.

**Decisional Diffie-Hellman problem (DDH):**
Given $g$ a generator, distinguish between the distribution $(g^a, g^b, g^{ab})$ and$(g^a, g^b, g^c)$ where $a$, $b$ and $c$ are uniformly sampled in $\mathbb{Z}_q$.

*DLP is the hardest to solve,*
*There exists groups where DDH is easy to solve but CDH difficult.*

# DDH experiment

Given an algorithm $\mathcal{G}$ which generates a group $\mathbb{G}$:

| $\mathcal{C}$ | | $\mathcal{B}$ |
|---|---|---|
| $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$<br>$x, y \leftarrow U(\mathbb{Z}_q)$ | | |
| RAND ($b = 0$): $z \leftarrow U(\mathbb{Z}_q)$<br>DDH ($b = 1$): $z = xy$ | $\xrightarrow{(g, g^x, g^y, g^z)}$ | |
| | | output $b'$ |

$$Adv(\mathcal{B}) = \left| \Pr[\mathcal{B} \xrightarrow{RAND} 1] - \Pr[\mathcal{B} \xrightarrow{DDH} 1] \right|.$$

**Definition**

The DDH problem is difficult to solve for a group $\mathcal{G}$ if for all Probabilistic Polynomial Time (PPT) algorithm $\mathcal{B}$, there exists a negligible function $negl(n)$ such that:

$$Adv(\mathcal{B}) \leq negl(n).$$

Probabilities are taken over the experiment in which $\mathcal{G}(1^n)$ outputs $(\mathbb{G}, q, g)$ then uniform $x, y, z \in \mathbb{Z}_q$ are chosen.

# Best known algorithm

- **Exhaustive search**: the security parameter $n$ is the number of bits of $q$ (the order of the group) i.e. $\log q = n$.
  Brute-force search: exponential in $n$.

- **Best known algorithm**: the general number field sieve (NFS).
  Complexity in $\exp(O(n)^{1/3}(\log n)^{2/3})$

# ElGamal encryption

- **Key generation:** generate $(\mathbb{G}, q, g)$ (publics), then sample an uniform $x$, and compute $h = g^x$, Secret key: $sk = x$, Public key: $pk = g^x$.
- **Enc:** given $pk = g^x$ and a message $m \in \mathbb{G}$, choose $y$ uniform and output $(g^y, h^y \cdot m)$.
- **Dec:** given $sk = x$ and $(c_1, c_2)$, output $m = c_2/c_1^x$.

**Correctness**: $\frac{c_2}{c_1^x} = \frac{h^y \cdot m}{g^{xy}} = \frac{g^{xy} \cdot m}{g^{xy}} = m$.

**Security**

If the DDH problem is difficult to solve, then the ElGamal encryption scheme is IND-CPA secure.

# Principe d'une preuve de sécurité

Security proof: **show that if an adversary can succeed in attacking the scheme with a non negligible advantage, then it is possible to solve a difficult problem (DDH).**

$\Rightarrow$ it's a reduction between two problems
Reduction from problem A to problem B


*show that if I know how to solve B then I know how to solve A:*
$\Rightarrow$ if A is difficult to solve, then so is B.

We start with an instance A (instance of DDH : $(g, g^x, g^y, g^z)$) and an oracle for B (which is the hypothesis that an adversary can successfully attack).

# Security proof

**Idea:** given the instance of DDH, we build one from the IND-CPA experiment and we use the answer of the adversary to solve DDH.

We recall the DDH experiment:

| $\mathcal{C}$ | | $\mathcal{B}$ |
|---|---|---|
| $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$<br>$x, y \leftarrow U(\mathbb{Z}_q)$ | | |
| RAND ($b = 0$): $z \leftarrow U(\mathbb{Z}_q)$<br>DDH ($b = 1$): $z = xy$ | $\xrightarrow{(g, g^x, g^y, g^z)}$ | |
| | | output $b'$ |

$$Adv(\mathcal{B}) = \left| \Pr[\mathcal{B} \xrightarrow{RAND} 1] - \Pr[\mathcal{B} \xrightarrow{DDH} 1] \right|.$$

# Security proof

**Idea:** given the instance of DDH, we build one from the IND-CPA experiment and we use the answer of the adversary to solve DDH.

We then recall the IND-CPA security game:

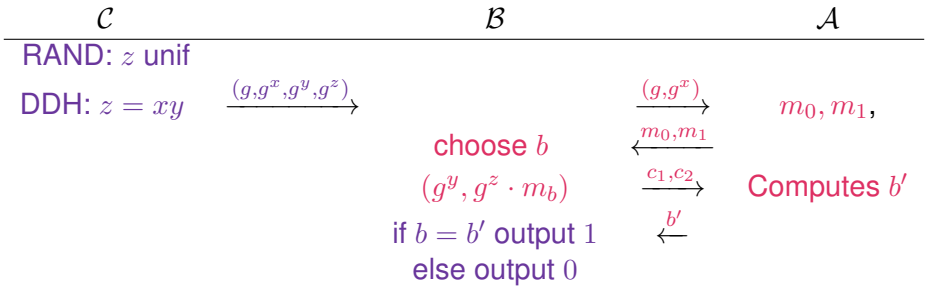
| $\mathcal{B}$ | | $\mathcal{A}$ |
|---|---|---|
| $(sk = x, pk = g^x) \leftarrow KeyGen(.)$ | $\xrightarrow{pk=g^x}$ | |
| choose $b$ | $\xleftarrow{m_0, m_1}$ | Chooses $m_0, m_1$, |
| computes $(c_1, c_2) \leftarrow Enc(pk, m_b)$ | $\xrightarrow{c_1, c_2}$ | Computes a bit $b'$ |
| | | if $b = b'$ then output Win |

We want to show that if DDH is difficult to solve, then there exists a negligible function $negl$ such that que:

$$\Pr[\mathcal{A} \text{ Win}] \leq 1/2 + negl(n).$$

# Security proof

**We have an algorithm $\mathcal{B}$ which wants to solve DDH using $\mathcal{A}$.**



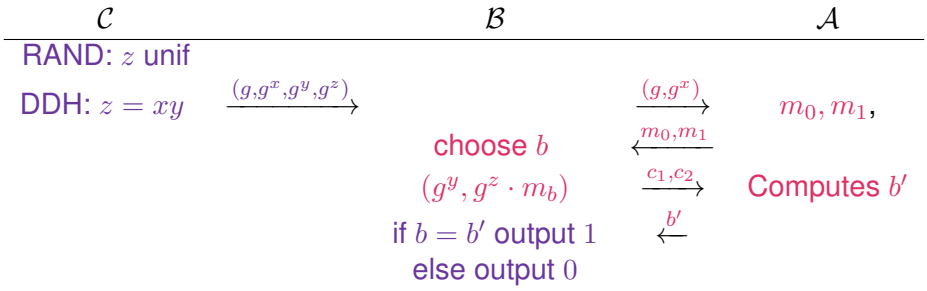

View of $\mathcal{B}$:

- If RAND: $z$ is uniformly distributed so $c_2$ too. $\mathcal{A}$ cannot distinguish between two ciphertexts: its advantage is zero, the probability that $\mathcal{B}$ outputs $1$ is then $1/2$.

$$\Pr[\mathcal{B} \xrightarrow{RAND} 1] = 1/2,$$

# Security proof

**We have an algorithm $\mathcal{B}$ which wants to solve DDH using $\mathcal{A}$.**




View of $\mathcal{B}$:

- If DDH: $z = xy$ and the ciphertext is exactly an ElGamal ciphertext. The probability that $\mathcal{B}$ outputs $1$ is exactly the success probability of $\mathcal{A}$ dans le in the IND-CPA security game (as it has the same view).

$$\Pr[\mathcal{B} \xrightarrow{DDH} 1] = \Pr[\mathcal{A} \text{ win}],$$

# Security proof

To conclude, we have:
$\Pr[\mathcal{B} \xrightarrow{RAND} 1] = 1/2$,
$\Pr[\mathcal{B} \xrightarrow{DDH} 1] = \Pr[\mathcal{A} \text{ win}]$,
donc :

$$\begin{aligned} Adv(\mathcal{B}) &= |\Pr[\mathcal{B} \xrightarrow{RAND} 1] - \Pr[\mathcal{B} \xrightarrow{DDH} 1]| \\ &= |\Pr[\mathcal{A} \text{ win}] - 1/2| \end{aligned}$$

Finally:

- As DDH is a difficult problem, we know there exists a negligible function $negl$ such that $Adv(\mathcal{B}) \leq negl$ then $\Pr[\mathcal{A} \text{ win}] \leq 1/2 + negl$.
- We suppose that $\mathcal{A}$ successfully attack, then there exists a non negligible $\varepsilon$ such that $\Pr[\mathcal{A} \text{ win}] \geq 1/2 + \varepsilon$, then $Adv(\mathcal{B}) \geq \varepsilon$ which implies there exists a distinguisher for the DDH problem.

# Today: an introduction to (lattice-based) cryptography

GREYC
Electronics and Computer Science Laboratory

1. El Gamal encryption scheme
   - Background
   - Discrete logarithm problem
   - El Gamal scheme and its security

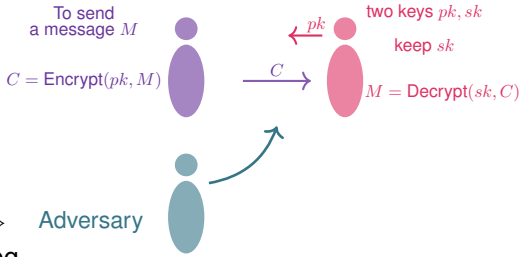
2. Regev's encryption scheme
   - Lattices and hard problem on lattices
   - Learning With Errors problem
   - How to encrypt using LWE?
   - Practical scheme using Module-LWE

# Post-quantum cryptography

Let's go back to the example: you want to send a message to someone.

Two possibilities:

- Either you share a secret key,
- Either you don't
  $\Rightarrow$ public key cryptography (RSA...).



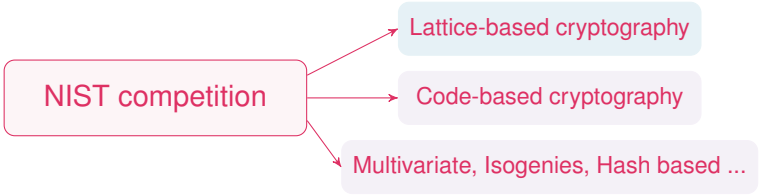

Solve a difficult algorithmic problem $\Leftrightarrow$
Examples: factorisation, discrete log

- Solving those problems needs an exponential complexity on a classical computer.
- Shor's algorithm (1995): **polynomial time on a quantum computer**.

# Context

→ need alternatives

- Post-quantum secure,
- Efficient,
- New functionalities, different types of constructions.



# NIST competition

From 2017 to 2024, NIST competition to develop new standards on post-quantum cryptography

Total: $69$ accepted submissions (round 1)

- Signature ($5$ lattice-based),
- Public key encryption / Key Encapsulation Mechanism ($21$ lattice-based)

**Other candidates:** $17$ code-based PKE, $7$ multivariate signatures, $3$ hash-based signatures, $7$ from "other" assumptions (isogenies, PQ RSA ...) and $4$ attacked + $5$ withdrawn.

$\Rightarrow$ lattice-based constructions are very serious candidates

**5 over 7 finalists** are lattice-based

2022 first results: **3 over 4 new standards** are lattice-based

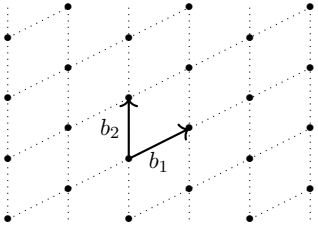
# Why lattice-based cryptography?

- Likely to resist attacks from quantum computers,
- Strong security guarantees,
  from well-understood hard problems on lattices.

- Novel and powerful cryptographic functionalities,
  - Public key encryption and signature scheme (practical),
  - Advanced signature (group signature ...),
    and encryption scheme (IBE, ABE, ...),
  - Fully homomorphic encryption.

- Efficiency

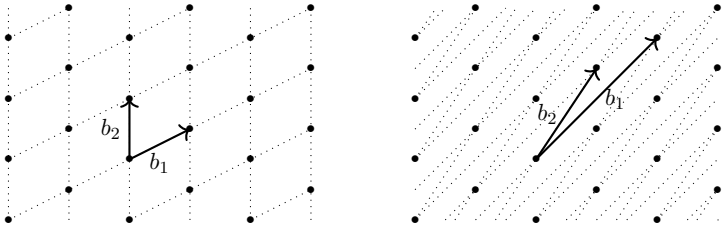
# Lattices




**Lattice**

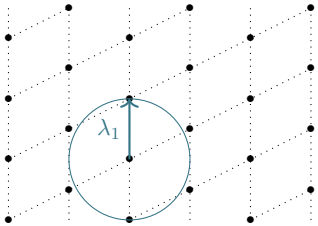
$\mathcal{L}(\mathbf{B}) = \{\sum_{1=i}^{n} a_i \mathbf{b}_i, a_i \in \mathbb{Z}\}$, where the $(\mathbf{b}_i)_{1 \leq i \leq n}$'s, linearly independent vectors, are a basis of $\mathcal{L}(\mathbf{B})$.

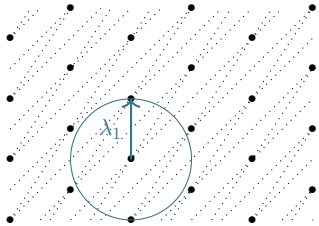
# Lattices




- Several basis define a lattice, some are better.

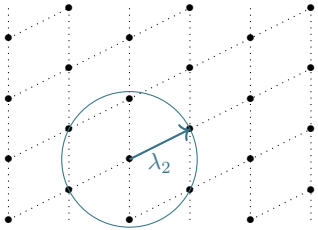
# Lattices

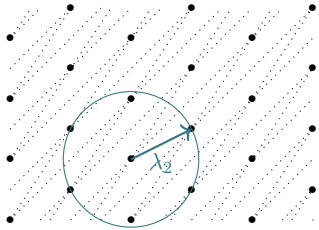



- Several basis define a lattice, some are better.
- The first minimum $\lambda_1$ is the norm of the smallest non-zero vector.

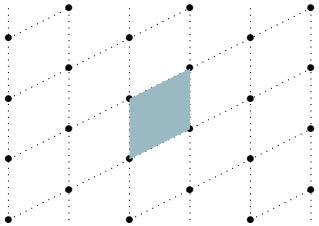
# Lattices

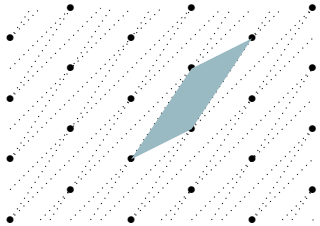



- Several basis define a lattice, some are better.
- The first minimum $\lambda_1$ is the norm of the smallest non-zero vector.
- The $n$-th minima $\lambda_n$ is the radius of a sphere which contains $n$ linearly independent shortest vectors of the lattices.

# Lattices





- Several basis define a lattice, some are better.
- The first minimum $\lambda_1$ is the norm of the smallest non-zero vector.
- The $n$-th minima $\lambda_n$ is the radius of a sphere which contains $n$ linearly independent shortest vectors of the lattices.
- The fundamental parallelepiped is defined by $\mathcal{P}(\mathbf{B}) = \{\sum_{i=1}^{n} c_i \mathbf{b}_i : c_i \in [0, 1)\}$. Its volume defines the volume of the lattice: $\det(\Lambda) = |\det(\mathbf{B})|$.

# Lattices

- The first minimum $\lambda_1$ is the norm of the smallest non-zero vector.
- The $n$-th minima $\lambda_n$ is the radius of a sphere which contains $n$ linearly independent shortest vectors of the lattices.
- The fundamental parallelepiped is defined by $\mathcal{P}(\mathbf{B}) = \{\sum_{i=1}^{n} c_i \mathbf{b}_i : c_i \in [0,1)\}$. Its volume defines the volume of the lattice: $\det(\Lambda) = |\det(\mathbf{B})|$.

- Minkowski Theorem:

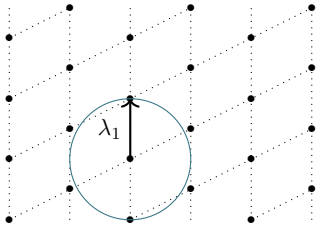
$$\lambda_1(\Lambda) \leq \sqrt{n} \cdot \det(\Lambda)^{1/n},$$

$$\left(\prod_{i=1}^{n} \lambda_i(\Lambda)\right)^{1/n} \leq \sqrt{n} \cdot \det(\Lambda)^{1/n}.$$

# Shortest Vector Problem (SVP)

Given a lattice $\mathcal{L}(\mathbf{B})$ of dimension $n$:

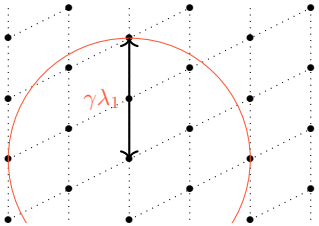
Output: find the shortest non-zero vector $\mathbf{x} \in \mathcal{L}(\mathbf{B})$.



# Approx Shortest Vector Problem (Approx SVP$_\gamma$)

Given a lattice $\mathcal{L}(\mathbf{B})$ of dimension $n$:

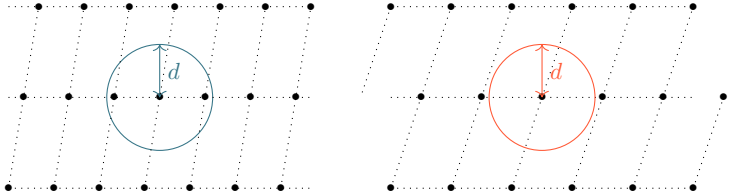
Output: find a non-zero vector $\mathbf{x} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{x}\| \leq \gamma\lambda_1(\mathcal{L}(\mathbf{B}))$



# Gap Shortest Vector Problem (GapSVP)

Given a lattice $\mathcal{L}(\mathbf{B})$ of dimension $n$ and $d > 0$:

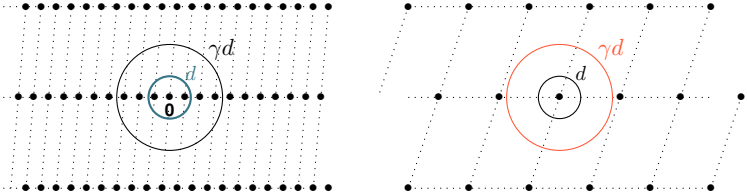
Output:
- YES: there is $\mathbf{z} \in \mathcal{L}(\mathbf{B})$ non-zero such that $\|\mathbf{z}\| < d$,
- NO: for all non-zero vectors $\mathbf{z} \in \mathcal{L}(\mathbf{B})$: $\|\mathbf{z}\| \geq d$.



# Gap Shortest Vector Problem (GapSVP$_\gamma$)

Given a lattice $\mathcal{L}(\mathbf{B})$ of dimension $n$ and $d > 0$:

Output:
- YES: there is $\mathbf{z} \in \mathcal{L}(\mathbf{B})$ non-zero such that $\|\mathbf{z}\| < d$,
- NO: for all non-zero vectors $\mathbf{z} \in \mathcal{L}(\mathbf{B})$: $\|\mathbf{z}\| \geq \gamma d$.

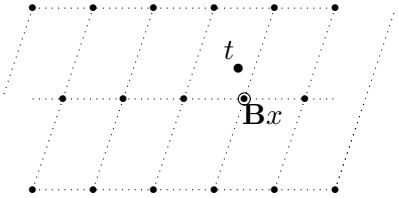


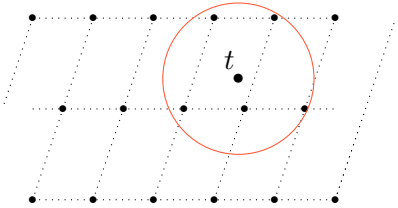
# Closest Vector Problem

Given a lattice $\mathcal{L}(\mathbf{B})$ of dimension $n$ and $\mathbf{t} \in \mathbb{Z}^m$:

Output: find $\mathbf{x} \in \mathbb{Z}^n$ minimizing $||\mathbf{Bx} - \mathbf{t}||$.

Approx variant: find $\mathbf{x} \in \mathbb{Z}^n$ such that $||\mathbf{Bx} - \mathbf{t}|| \leq \gamma \cdot \text{dist}(\mathbf{t}, \Lambda(\mathbf{B}))$.



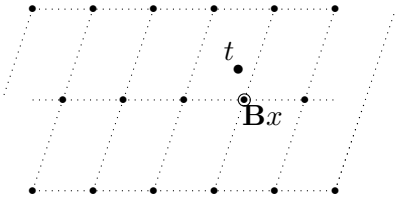



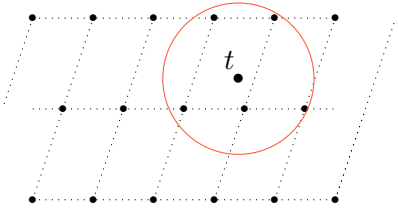
# Closest Vector Problem

Given a lattice $\mathcal{L}(\mathbf{B})$ of dimension $n$ and $\mathbf{t} \in \mathbb{Z}^m$:

Output: find $\mathbf{x} \in \mathbb{Z}^n$ minimizing $||\mathbf{Bx} - \mathbf{t}||$.

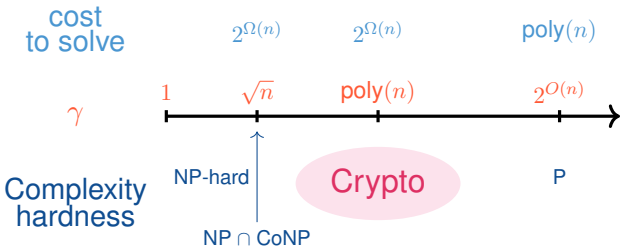
Approx variant: find $\mathbf{x} \in \mathbb{Z}^n$ such that $||\mathbf{Bx} - \mathbf{t}|| \leq \gamma \cdot \text{dist}(\mathbf{t}, \Lambda(\mathbf{B}))$.




How hard is it to solve those problems?

# Hardness of Approx SVP$_\gamma$




**Conjecture**

There is no polynomial time algorithm that approximates this lattice problem and its variants to within polynomial factors.

# At the heart of lattice-based cryptography the Learning With Errors problem

- Introduced by Regev in 2005

**Problem**: solve a linear system with $m$ equations and $n$ variables ($m \geq n$), with noise, and modulo an integer $q$.

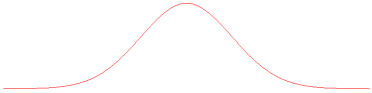
Find $(s_1, s_2, s_3, s_4, s_5)$ such that:

$$
\begin{aligned}
s_1 + 22s_2 + 17s_3 + 2s_4 + s_5 &\approx 16 \mod 23 \\
3s_1 + 2s_2 + 11s_3 + 7s_4 + 8s_5 &\approx 17 \mod 23 \\
15s_1 + 13s_2 + 10s_3 + 3s_4 + 5s_5 &\approx 3 \mod 23 \\
17s_1 + 11s_2 + 20s_3 + 9s_4 + 3s_5 &\approx 8 \mod 23 \\
2s_1 + 14s_2 + 13s_3 + 6s_4 + 7s_5 &\approx 9 \mod 23 \\
4s_1 + 21s_2 + 9s_3 + 5s_4 + s_5 &\approx 18 \mod 23 \\
11s_1 + 12s_2 + 5s_3 + s_4 + 9s_5 &\approx 7 \mod 23
\end{aligned}
$$

# Gaussian distributions

Continuous Gaussian distribution of center $c$ and parameter $s$:

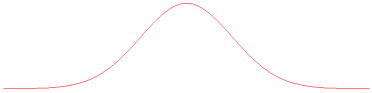
$$D_{s,c}(x) \sim \frac{1}{s} \exp\left( -\pi \frac{||x-c||^2}{s^2} \right)$$
$$\forall x \in \mathbb{R}$$



# Gaussian distributions

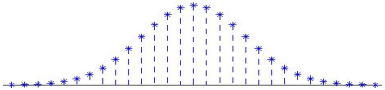
Continuous Gaussian distribution of center $c$ and parameter $s$:

$$D_{s,c}(x) \sim \frac{1}{s} \exp\left(-\pi \frac{||x-c||^2}{s^2}\right)$$
$$\forall x \in \mathbb{R}$$



Gaussian distribution on $\mathbb{Z}$ of center $c$ with parameter $s$:

$$D_{\mathbb{Z},s,c}(x) \sim \frac{1}{s} \exp\left(-\pi \frac{||x-c||^2}{s^2}\right)$$
$$\forall x \in \mathbb{Z}$$



- It is not the rounding of the continuous Gaussian.
- We now how to sample it efficiently.
- Almost all samples are in $[-t \cdot s, +t \cdot s]$ for a constant $t$, if $s$ is not to small.

# Discrete gaussian on lattices

**Theorem (Gentry, Peikert, Vaikuntanathan 2008)**

*There exists a PPT algorithm which, given a basis $\mathbf{B}$ of a lattice $\Lambda(\mathbf{B})$ of dimension $n$, a parameter $s \geq \|\tilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log n})$, an a center $c \in \mathbb{R}^n$, outputs a sample from a distribution statistically close from $D_{\Lambda,s,c}$.*

Intuition: sampling on $\mathbb{Z}$ is quite easy, it is more complicated on a general lattice.

**Important:** Better is the basis (with short vectors), smaller is the parameter we can sample with, and then have short vectors.

# The Learning With Errors problem [Regev 05]

Let $n > 1$, $q \geq 2$ and $\alpha \in ]0, 1[$.
For any $\mathbf{s} \in \mathbb{Z}_q^n$, we define the distribution $\mathcal{D}_{n,q,\alpha}(\mathbf{s})$ by:

$$(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e), \text{ with } \mathbf{a} \leftarrow U(\mathbb{Z}_q^n) \text{ and } e \leftarrow D_{\mathbb{Z},\alpha q}.$$

- Search LWE
  For any $\mathbf{s}$: find $\mathbf{s}$ given an arbitrary number of samples from $\mathcal{D}_{n,q,\alpha}(\mathbf{s})$.

- Decision LWE
  With non-negligible probability on $\mathbf{s} \leftarrow U(\mathbb{Z}_q^n)$: distinguish between the distributions $\mathcal{D}_{n,q,\alpha}(\mathbf{s})$ and $U(\mathbb{Z}_q^{n+1})$.

# Decision version

Let $n > 1$, $q \geq 2$ and $\alpha \in ]0, 1[$.
For any $\mathbf{s} \in \mathbb{Z}_q^n$, we define the distribution $\mathcal{D}_{n,q,\alpha}(\mathbf{s})$ by:

$$(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e), \text{ with } \mathbf{a} \leftarrow U(\mathbb{Z}_q^n) \text{ and } e \leftarrow D_{\mathbb{Z},\alpha q}.$$

- Decision LWE
  With non-negligible probability on $\mathbf{s} \leftarrow U(\mathbb{Z}_q^n)$: distinguish between the distributions $\mathcal{D}_{n,q,\alpha}(\mathbf{s})$ and $U(\mathbb{Z}_q^{n+1})$.

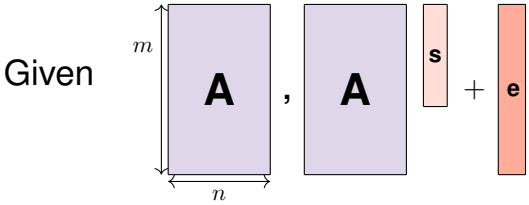
We consider an oracle $\mathcal{O}$ which produces independant samples, all from the same distribution being:

- either $\mathcal{D}_{n,q,\alpha}(\mathbf{s})$ for a fixed $\mathbf{s}$,
- either $U(\mathbb{Z}_q^{n+1})$.

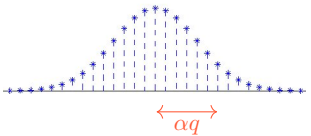
The goal is to decide which one with a non-negligeable advantage.

# The Learning With Errors problem

$\mathbf{LWE}^{n}_{\alpha,q}$




- $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m \times n})$,
- $\mathbf{s} \leftarrow U(\mathbb{Z}_q^{n})$,
- $\mathbf{e} \leftarrow D_{\mathbb{Z}^m, \alpha q}$, small compared to $q$.




Discrete Gaussian error $D_{\mathbb{Z}, \alpha q}$

Search version: Given $(\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e})$, find $\mathbf{s}$.
Decision version: Distinguish from $(\mathbf{A}, \mathbf{b})$ with $\mathbf{b}$ uniform.

# Equivalence between the two variants

LWE sample: $(\mathbf{A}, \mathbf{b} = \mathbf{A}\,\mathbf{s} + \mathbf{e} \bmod q)$ with short $\mathbf{e}$.

- Easy reduction : from **decision** to **search**
  - find **s** $\Rightarrow$ distinguish **b** uniform or **b** LWE sample,

# Equivalence between the two variants

LWE sample: $(\mathbf{A}, \mathbf{b} = \mathbf{A}\,\mathbf{s} + \mathbf{e} \bmod q)$ with short $\mathbf{e}$.

- Easy reduction : from **decision** to **search**
  - find $\mathbf{s} \Rightarrow$ distinguish $\mathbf{b}$ uniform or $\mathbf{b}$ LWE sample,
  - Given $(\mathbf{A}, \mathbf{b})$, find the oracle to find $\mathbf{s}$, compute $\mathbf{b} - \mathbf{As}$:

# Equivalence between the two variants

LWE sample: $(\mathbf{A}, \mathbf{b} = \mathbf{A}\,\mathbf{s} + \mathbf{e} \bmod q)$ with short $\mathbf{e}$.

- Easy reduction : from **decision** to **search**
  - find $\mathbf{s} \Rightarrow$ distinguish $\mathbf{b}$ uniform or $\mathbf{b}$ LWE sample,
  - Given $(\mathbf{A}, \mathbf{b})$, find the oracle to find $\mathbf{s}$, compute $\mathbf{b} - \mathbf{As}$:
    - if it is small, then $\mathbf{b}$ is an LWE sample,
    - if it looks uniform, then $\mathbf{b}$ is uniform.

# Equivalence between the two variants

LWE sample: $(\mathbf{A}, \mathbf{b} = \mathbf{A}\,\mathbf{s} + \mathbf{e} \bmod q)$ with short $\mathbf{e}$.

- Easy reduction : from **decision** to **search**
  - find **s** $\Rightarrow$ distinguish **b** uniform or **b** LWE sample,
  - Given $(\mathbf{A}, \mathbf{b})$, find the oracle to find **s**, compute $\mathbf{b} - \mathbf{As}$:
    - if it is small, then **b** is an LWE sample,
    - if it looks uniform, then **b** is uniform.
- 2nd reduction: from **search** to **decision**
  - Distinguish **b** uniform from **b** LWE sample $\Rightarrow$ find **s**,

# Equivalence between the two variants

LWE sample: $(\mathbf{A}, \mathbf{b} = \mathbf{A}\,\mathbf{s} + \mathbf{e} \bmod q)$ with short $\mathbf{e}$.

- Easy reduction : from **decision** to **search**
  - find **s** $\Rightarrow$ distinguish **b** uniform or **b** LWE sample,
  - Given $(\mathbf{A}, \mathbf{b})$, find the oracle to find **s**, compute $\mathbf{b} - \mathbf{As}$:
    - if it is small, then **b** is an LWE sample,
    - if it looks uniform, then **b** is uniform.

- 2nd reduction: from **search** to **decision**
  - Distinguish **b** uniform from **b** LWE sample $\Rightarrow$ find **s**,
  - Given $(\mathbf{A}, \mathbf{b})$ use the oracle to find each coordinate of **s**: for all $s_1^*$, choose $u$ uniform in $\mathbb{Z}_q$ and modify $(\mathbf{A}, \mathbf{b})$ as follow:

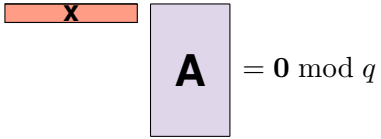
$$(\mathbf{a}, b) + (u, 0, \ldots, 0, us_1^*) = (\mathbf{a}', \langle \mathbf{a}', \mathbf{s} \rangle + e + u(s_1^* - s_1)), .$$

    - if $s_1^* = s_1$ it stays a LWE sample,
    - else **b** will be uniform.

# Short Integer Solution problem [Ajtai 1996]

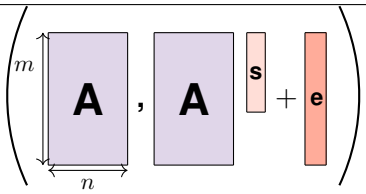
For $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m \times n})$:

## $\mathrm{SIS}_\beta$




Goal: Given $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m \times n})$, find $\mathbf{x}$ s.t. $0 < \|\mathbf{x}\| \leq \beta$.

## $\mathrm{LWE}_\alpha$



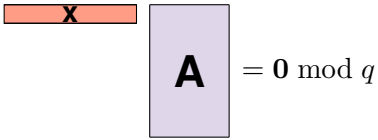

$\mathbf{s} \leftarrow U(\mathbb{Z}_q^n)$, $\mathbf{e} \leftarrow D_{\mathbb{Z}^m, \alpha q}$.

Goal: Given $(\mathbf{A}, \mathbf{A}\,\mathbf{s} + \mathbf{e})$, find $\mathbf{s}$.

# Short Integer Solution problem [Ajtai 1996]

For $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m \times n})$:

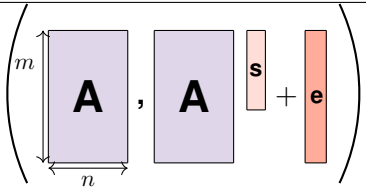
## $\mathrm{SIS}_\beta$




Solve SVP in

$$\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m | \mathbf{x}^T \mathbf{A} = 0 \bmod q\}$$

## $\mathrm{LWE}_\alpha$




$\mathbf{s} \leftarrow U(\mathbb{Z}_q^n)$, $\mathbf{e} \leftarrow D_{\mathbb{Z}^m, \alpha q}$.

Solve CVP in

$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y} = \mathbf{A}\,\mathbf{s} \bmod q \text{ for some } \mathbf{s} \in \mathbb{Z}^n\}$$

# Hardness of LWE

- Exhaustive search
  - Try all the $\mathbf{s} \in \mathbb{Z}_q^n \rightarrow$ is $\mathbf{b} - \mathbf{As}$ small?
  - $\Rightarrow$ cost around $q^n$.

# Hardness of LWE

- Exhaustive search
  - Try all the $\mathbf{s} \in \mathbb{Z}_q^n \rightarrow$ is $\mathbf{b} - \mathbf{As}$ small?
  - $\Rightarrow$ cost around $q^n$.
  - Other possibility: guess the $n$ first errors, find $\mathbf{s} \rightarrow$ is $\mathbf{b} - \mathbf{As}$ small?
  - $\Rightarrow$ cost around $(\alpha q \sqrt{n})^n$.

# Hardness of LWE

- Exhaustive search
  - Try all the $\mathbf{s} \in \mathbb{Z}_q^n \rightarrow$ is $\mathbf{b} - \mathbf{As}$ small?
  - $\Rightarrow$ cost around $q^n$.
  - Other possibility: guess the $n$ first errors, find $\mathbf{s} \rightarrow$ is $\mathbf{b} - \mathbf{As}$ small?
  - $\Rightarrow$ cost around $(\alpha q \sqrt{n})^n$.

- How to do better?
  - LWE is a lattice problem: consider

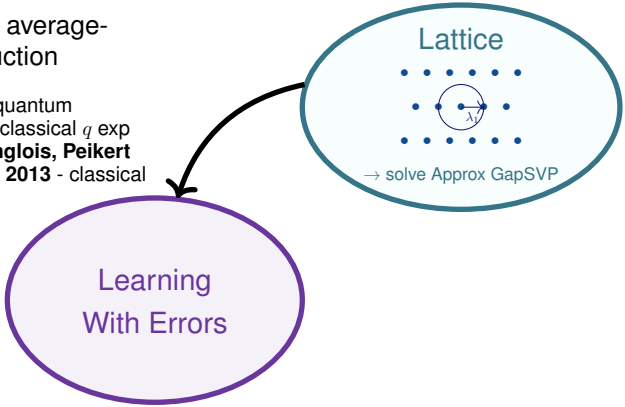
$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y} = \mathbf{As} \bmod q \text{ for } \mathbf{s} \in \mathbb{Z}^n\}.$$

    Solving LWE $\Leftrightarrow$ solving CVP in this lattice.
  - Cost: $\left(\frac{n \log q}{\log^2 \alpha}\right)^{\frac{n \log q}{\log^2 \alpha}}$.

# Hardness of the Learning With Errors problem

Worst-case to average-case reduction

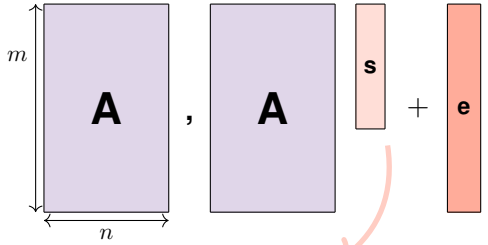
- **Regev 2005** - quantum
- **Peikert 2009** - classical $q$ exp
- **Brakerski, Langlois, Peikert Regev, Stehlé 2013** - classical



# LWE variants

Choose another distribution for the secret or the error.

Regev 2009: uniform secret and gaussian error.



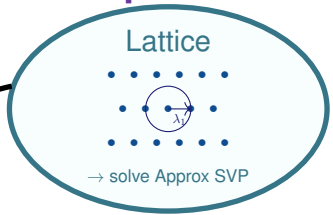

- Gaussian (continue, discretize, discrete ...),
- Uniform in small interval,
- Binary under conditions.

- Same distribution as the error: in particular Gaussian,
- Binary (Unif in $\{0,1\}^n$),
- Entropic.

# Hardness of the Learning With Errors problem

Worst-case to average-case reduction

- **Regev 2005** - quantum
- **Peikert 2009** - classical $q$ exp
- **Brakerski, Langlois, Peikert Regev, Stehlé 2013** - classical



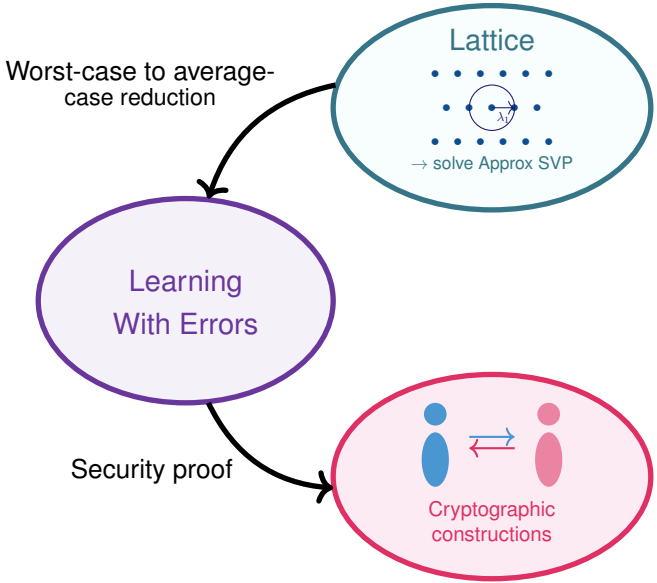

Learning With Errors

- **Peikert 2010** - discrete Gaussian noise
- **Döttling, Müller-Quade 2013** - small uniform
- **Micciancio, Peikert 2013** - small uniform and binary noise
- **Bai, Lepoint, Roux-Langlois, Sakzad, Stehlé, Steinfeld 2015** - small uniform, dimension preserving

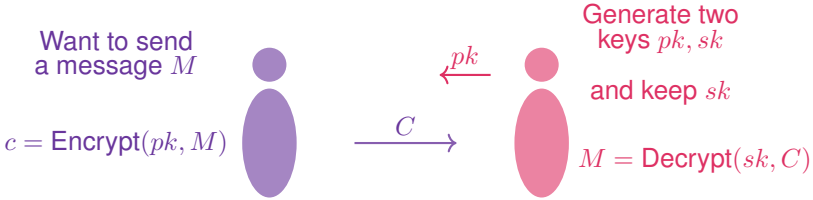
Self reductions

- **Applebaum, Cash, Peikert, Sahai 2009** - same error and secret
- **Goldwasser, Kalai, Peikert, Vaikuntanathan 2010** - binary secret
- **Brakerski, Langlois, Peikert, Regev, Stehlé 2013** - binary secret
- **Micciancio 2018** - binary secret
- **Brakerski, Döttling 2020** - entropic secret

# Using LWE to build provable constructions - theory



# Public key encryption

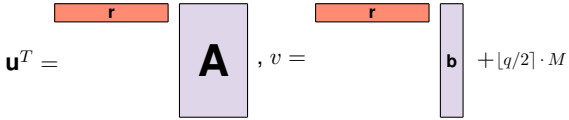


# Regev's encryption scheme

- **Parameters**: $n, m, q \in \mathbb{Z}$, $\alpha \in \mathbb{R}$,
- **Keys**: sk = $\mathbf{s}$ and pk = ( $\mathbf{A}$, $\mathbf{b}$ ), with $\mathbf{b} = \mathbf{A}\,\mathbf{s} + \mathbf{e} \bmod q$ where $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m \times n})$, $\mathbf{e} \hookleftarrow D_{\mathbb{Z}^m, \alpha q}$.

# Regev's encryption scheme

- **Parameters**: $n, m, q \in \mathbb{Z}$, $\alpha \in \mathbb{R}$,
- **Keys**: sk = $\mathbf{s}$ and pk = ($\mathbf{A}$, $\mathbf{b}$), with $\mathbf{b} = \mathbf{A}\,\mathbf{s} + \mathbf{e} \bmod q$
  where $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m \times n})$, $\mathbf{e} \hookleftarrow D_{\mathbb{Z}^m, \alpha q}$.
- **Encryption** $(M \in \{0,1\})$**:** Let $\mathbf{r} \hookleftarrow U(\{0,1\}^m)$,



# Regev's encryption scheme

- **Parameters**: $n, m, q \in \mathbb{Z}$, $\alpha \in \mathbb{R}$,
- **Keys**: sk = $\mathbf{s}$ and pk = ($\mathbf{A}$, $\mathbf{b}$), with $\mathbf{b} = \mathbf{A}\,\mathbf{s} + \mathbf{e} \bmod q$
  where $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m \times n})$, $\mathbf{e} \hookleftarrow D_{\mathbb{Z}^m, \alpha q}$.
- **Encryption** ($M \in \{0,1\}$)**:** Let $\mathbf{r} \hookleftarrow U(\{0,1\}^m)$,

$$\mathbf{u}^T = \mathbf{r}^T\mathbf{A}, \quad v = \mathbf{r}^T\mathbf{b} + \lfloor q/2 \rceil \cdot M$$

- **Decryption** of $(\mathbf{u}, v)$: compute $v - \mathbf{u}^T\mathbf{s}$,

$$\underbrace{\mathbf{r}^T\left[\mathbf{A}\,\mathbf{s} + \mathbf{e}\right] + \lfloor q/2 \rceil \cdot M}_{v} - \underbrace{\mathbf{r}^T\mathbf{A}\,\mathbf{s}}_{\mathbf{u}^T\mathbf{s}} = \text{small} + \lfloor q/2 \rceil \cdot M$$

If **close from** $0$: return $0$, if **close from** $\lfloor q/2 \rfloor$: return $1$.

# Regev's encryption scheme

- **Parameters**: $n, m, q \in \mathbb{Z}$, $\alpha \in \mathbb{R}$,
- **Keys**: sk = $\mathbf{s}$ and pk = ($\mathbf{A}$, $\mathbf{b}$), with $\mathbf{b} = \mathbf{A}\,\mathbf{s} + \mathbf{e} \bmod q$
  where $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m \times n})$, $\mathbf{e} \hookleftarrow D_{\mathbb{Z}^m, \alpha q}$.
- **Encryption** $(M \in \{0,1\})$**:** Let $\mathbf{r} \hookleftarrow U(\{0,1\}^m)$,

$$\mathbf{u}^T = \mathbf{r}^T \mathbf{A}, \quad v = \mathbf{r}^T \mathbf{b} + \lfloor q/2 \rceil \cdot M$$

- **Decryption** of $(\mathbf{u}, v)$: compute $v - \mathbf{u}^T\mathbf{s}$,

$$\mathbf{r}^T \left[\mathbf{A}\,\mathbf{s} + \mathbf{e}\right] + \lfloor q/2 \rceil \cdot M - \mathbf{r}^T \mathbf{A}\,\mathbf{s} = \text{small} + \lfloor q/2 \rceil \cdot M$$

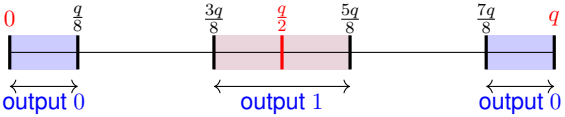
**LWE hard $\Rightarrow$ Regev's scheme is IND-CPA secure.**

# Correction

The randomness **r** is uniformly chosen in $\{0,1\}^m$,
and **e** is sampled from a discrete gaussian of parameter $\alpha q \leq q/(8m)$,
then, with overwhelming probability,

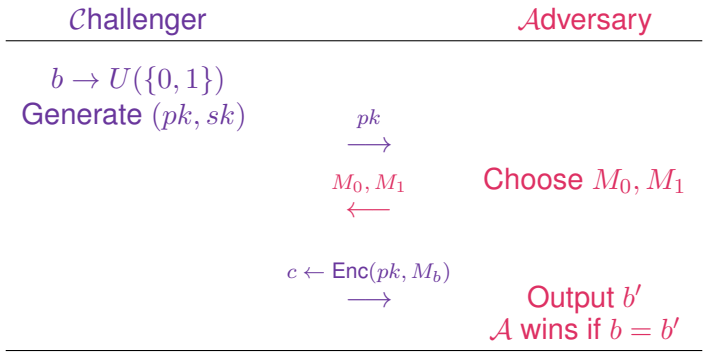
$$\Big|\sum_{i\leq m} r_i e_i\Big| \leq \|\mathbf{r}\| \cdot \|\mathbf{e}\| \leq \sqrt{m} \cdot \frac{q}{8\sqrt{m}} = \frac{q}{8}.$$

$v - \mathbf{u}^T\mathbf{s}$ is either close from $0$, either close from $\lfloor q/2 \rceil$, which allows to find $M$.



# IND-CPA security

To define the security, we use a game between a challenger and an adversary. We define the following experiment:




$$Adv^{CPA}(\mathcal{A}) = |\Pr[\mathcal{A} \text{ wins}] - 1/2|$$

# IND-CPA security

**Goal of the proof:** show that if an adversary succeed in attacking the encryption scheme with a non-negligible advantage, then the challenger can use it to solve a difficult problem (here LWE).

Decision LWE can also be seen as a game:

| $\mathcal{C}$ | | $\mathcal{B}$ |
|---|---|---|
| $\mathbf{A} \longleftarrow U(\mathbb{Z}_q^{m \times n})$ | | |
| RAND $(b = 0)$: $\mathbf{b} \leftarrow U(\mathbb{Z}_q^m)$ | | |
| LWE $(b = 1)$: $\mathbf{b} = \mathbf{As} + \mathbf{e}$ | $\xrightarrow{(\mathbf{A},\mathbf{b})}$ | |
| | | output $b'$ |

$$Adv(\mathcal{B}) = \left| \Pr[\mathcal{B} \xrightarrow{RAND} 1] - \Pr[\mathcal{B} \xrightarrow{LWE} 1] \right|.$$

# Leftover Hash Lemma

Let $m, n, q \geq 1$ be integers such that $m \geq 4n \log q$ and $q$ prime, and let $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m \times n})$ and $\mathbf{r} \hookleftarrow U(\{0,1\}^m)$. Then $(\mathbf{A}, \mathbf{r}^T\mathbf{A})$ has statistical distance $\leq 2^{-n}$ from the uniform distribution on $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^n$.

# Leftover Hash Lemma

Let $m, n, q \geq 1$ be integers such that $m \geq 4n \log q$ and $q$ prime, and let $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m \times n})$ and $\mathbf{r} \hookleftarrow U(\{0,1\}^m)$. Then $(\mathbf{A}, \mathbf{r}^T\mathbf{A})$ has statistical distance $\leq 2^{-n}$ from the uniform distribution on $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^n$.

- Statistical distance : $\Delta(D_1, D_2) = \frac{1}{2} \sum_x |D_1(x) - D_2(x)|$.

# Leftover Hash Lemma

Let $m, n, q \geq 1$ be integers such that $m \geq 4n \log q$ and $q$ prime, and let $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m \times n})$ and $\mathbf{r} \hookleftarrow U(\{0,1\}^m)$. Then $(\mathbf{A}, \mathbf{r}^T\mathbf{A})$ has statistical distance $\leq 2^{-n}$ from the uniform distribution on $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^n$.

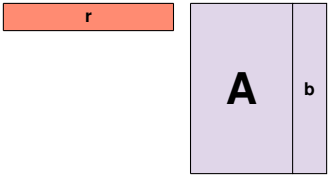
- Statistical distance : $\Delta(D_1, D_2) = \frac{1}{2} \sum_x |D_1(x) - D_2(x)|$.
- For any algorithm $\mathcal{A}$, we have
  $|\Pr[\mathcal{A}(D_1) = 1] - \Pr[\mathcal{A}(D_2) = 1]| \leq \Delta(D_1, D_2)$.
  $\Delta(D_1, D_2)$ small $\Rightarrow$ $D_1$ and $D_2$ are statistically indistinguishable.

# Leftover Hash Lemma

Let $m, n, q \geq 1$ be integers such that $m \geq 4n \log q$ and $q$ prime, and let $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m \times n})$ and $\mathbf{r} \hookleftarrow U(\{0,1\}^m)$. Then $(\mathbf{A}, \mathbf{r}^T\mathbf{A})$ has statistical distance $\leq 2^{-n}$ from the uniform distribution on $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^n$.

- Statistical distance : $\Delta(D_1, D_2) = \frac{1}{2} \sum_x |D_1(x) - D_2(x)|$.
- For any algorithm $\mathcal{A}$, we have
  $|\Pr[\mathcal{A}(D_1) = 1] - \Pr[\mathcal{A}(D_2) = 1]| \leq \Delta(D_1, D_2)$.
  $\Delta(D_1, D_2)$ small $\Rightarrow$ $D_1$ and $D_2$ are statistically indistinguishable.

The LHL implies that ( **(A b)** , **r** **(A b)** ) is indistinguishable from uniform.



# IND-CPA security proof

**Idea:** we start from an LWE instance, and build an instance of the IND-CPA experiment, then we use the answer of the adversary to solve LWE.
We use the following IND-CPA game:

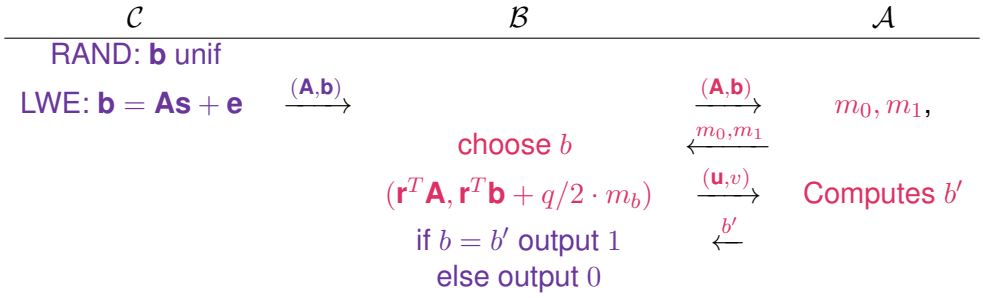
| $\mathcal{B}$ | | $\mathcal{A}$ |
|---|---|---|
| $(sk = \mathbf{s}, pk = (\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e}) \leftarrow KeyGen(.)$ | $\xrightarrow{pk=(\mathbf{A},\mathbf{b})}$ | |
| chooses $b$ | $\xleftarrow{m_0,m_1}$ | Chooses $m_0, m_1$, |
| computes $(\mathbf{u}, v) \leftarrow Enc(pk, m_b)$ | $\xrightarrow{(\mathbf{u},v)}$ | Computes a bit $b'$ |
| | | if $b = b'$ then output Win |

We want to show that if LWE is hard, then there exists a negligible function $negl$ such that:

$$\Pr[\mathcal{A} \text{ Win}] \leq 1/2 + negl(n).$$

# IND-CPA security proof

$\mathcal{B}$ **wants to solve decisional LWE using** $\mathcal{A}$**.**



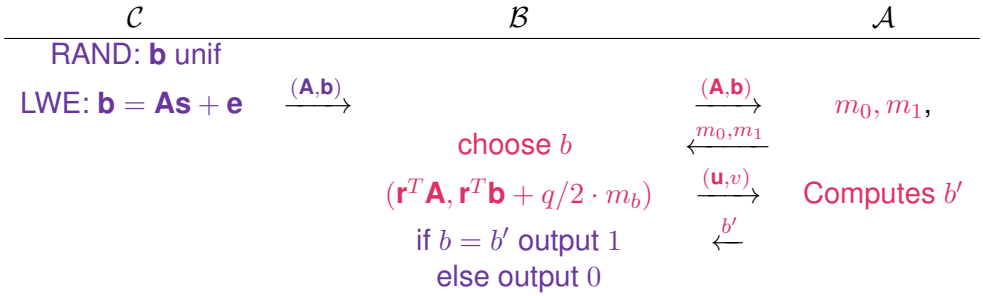

For $\mathcal{B}$:

- RAND: **b** is uniform then $v$ is uniform. $\mathcal{A}$ cannot distinguish between the two cases, its advantage is equals to zero, the probability that $\mathcal{B}$ outputs $1$ is $1/2$.

$$\Pr[\mathcal{B} \xrightarrow{RAND} 1] = 1/2,$$

# IND-CPA security proof

$\mathcal{B}$ wants to solve decisional LWE using $\mathcal{A}$.




For $\mathcal{B}$:

- LWE: $\mathbf{b} = \mathbf{As} + \mathbf{e}$ and then the ciphertext is exactly a ciphertext from the Regev encryption scheme. The probability that $\mathcal{B}$ outputs $1$ is exactly the success probability of $\mathcal{A}$ in the encryption scheme security game (as it has the same view of the experiment).

$$\Pr[\mathcal{B} \xrightarrow{LWE} 1] = \Pr[\mathcal{A} \text{ win}],$$

# IND-CPA security proof

To conclude, we have:

$$\Pr[\mathcal{B} \xrightarrow{RAND} 1] = 1/2,$$

$$\Pr[\mathcal{B} \xrightarrow{LWE} 1] = \Pr[\mathcal{A} \text{ win}],$$

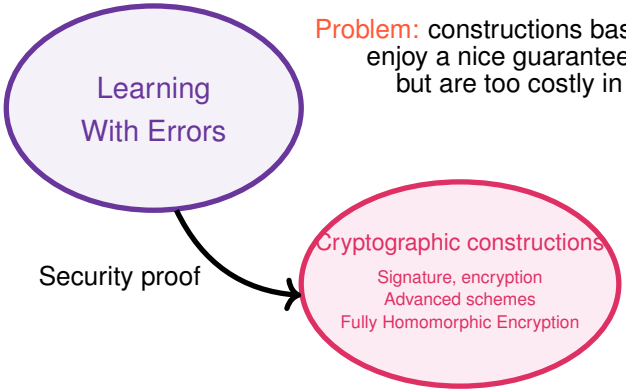
then:

$$\begin{aligned} Adv(\mathcal{B}) &= |\Pr[\mathcal{B} \xrightarrow{RAND} 1] - \Pr[\mathcal{B} \xrightarrow{LWE} 1]| \\ &= |\Pr[\mathcal{A} \text{ win}] - 1/2| \end{aligned}$$

If $\mathcal{A}$ succeeds with a non-negligible probability, then there exists $\varepsilon$ such that $\Pr[\mathcal{A} \text{ win}] \geq 1/2 + \varepsilon$, then $Adv(\mathcal{B}) \geq \varepsilon$ which implies that there exists a distinguisher able to solve the decisional LWE problem.

# Using LWE

Hardness of SIS/LWE used as a foundation for many constructions.




Problem: constructions based on SIS/LWE enjoy a nice guarantee of security but are too costly in practice.

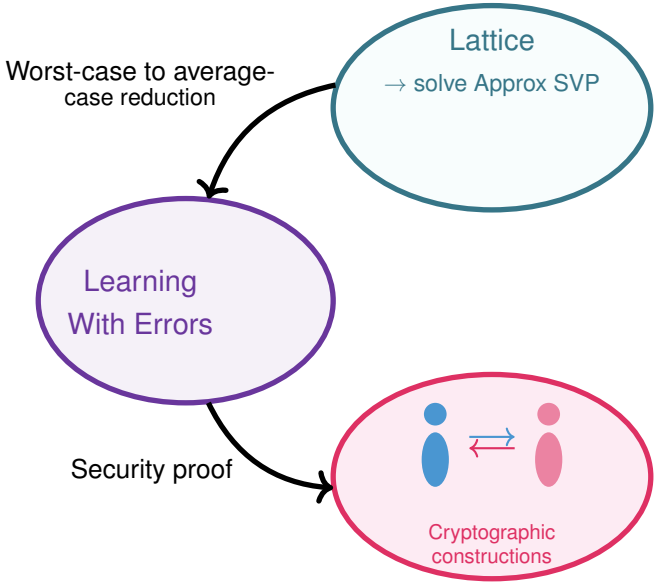
Solutions used today?

# Lattice-based NIST finalists

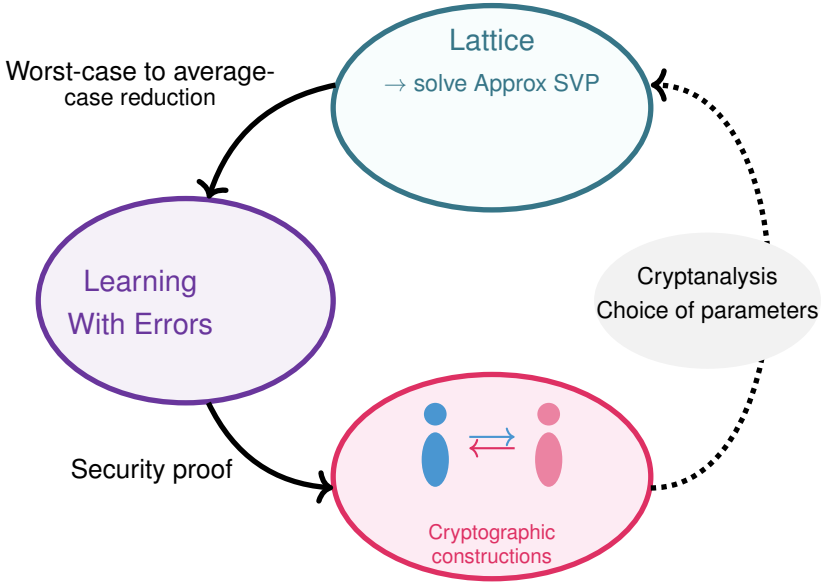
Among the 5 lattice-based finalists, 3 of them are based on (possibly structured) variants of LWE.

- Public Key Encryption
  - **Crystals - Kyber**: Module-LWE with both secret and noise chosen from a centered binomial distribution.
  - **Saber:** Module-LWR (deterministic variant).
  - **NTRU**
  - **FrodoKEM** (as alternate candidate): LWE but with smaller parameters.
- Signature
  - **Crystals - Dilithium**: Module-LWE with both secret and noise chosen in a small uniform interval, and Module-SIS.
  - **Falcon**: Ring-SIS on NTRU matrices.

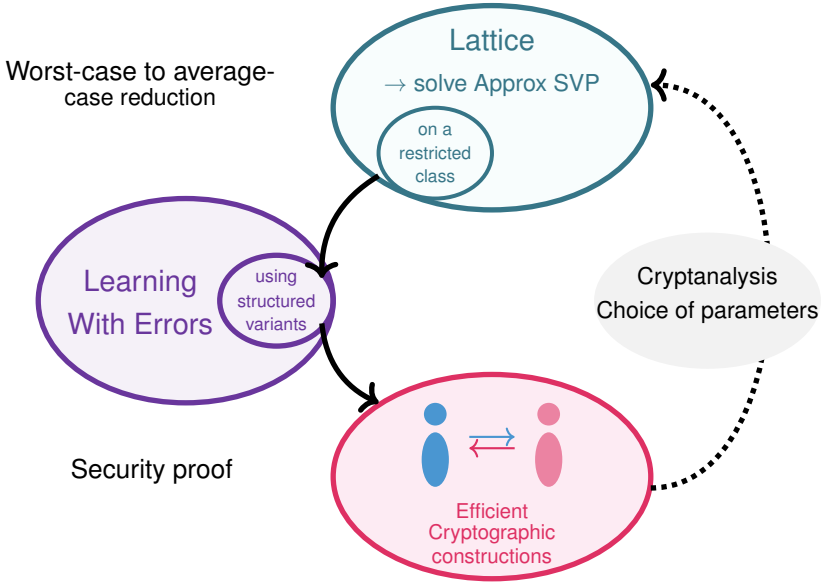
# Using SIS/LWE to build constructions



# Using SIS/LWE to build constructions in practice



# Using SIS/LWE to build constructions in practice

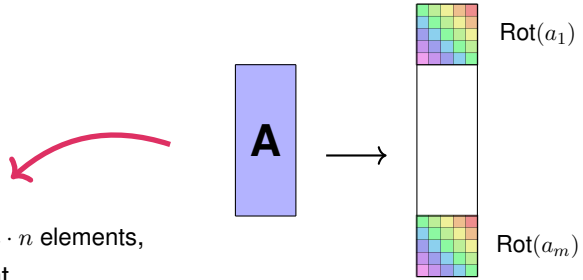


# From SIS/LWE to structured variants

**Problem:** constructions based on LWE enjoy a nice guaranty of security but are too costly in practice.

→ replace $\mathbb{Z}^n$ by a Ring, for example $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ ($n = 2^k$).

- Ring variants since 2006:




- Structured $\mathbf{A} \in \mathbb{Z}_q^{m \cdot n \times n}$ represented by $m \cdot n$ elements,
- Product with matrix/vector more efficient,
- Hardness of Ring-SIS, [Lyubashevsky and Micciancio 06] and [Peikert and Rosen 06]
- Hardness of Ring-LWE [Lyubashevsky, Peikert and Regev 10].

# Idea: replace $\mathbb{Z}^n$ by $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$

where $n = 2^k$ then the polynomial $x^n + 1$ is irreducible.
Elements of this ring are polynomials of degree less than $n$.

$R$ **is isomorph to** $\mathbb{Z}^n$

Let $a \in R$, we have $a(x) = a_0 + a_1 x + \ldots + a_{n-1}x^{n-1}$,
the isomorphism $R \to \mathbb{Z}^n$ associate

the polynomial $a \in R$ to the vector $\mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} \in \mathbb{Z}^n$.

# Idea: replace $\mathbb{Z}^n$ by $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$

Let's look at the product of two polynomials $x^n + 1$

- $a(x) = a_0 + a_1 \cdot x + \ldots + a_{n-1} \cdot x^{n-1}$
- $s(x) = s_0 + a_1 \cdot x + \ldots + a_{n-1} \cdot x^{n-1}$

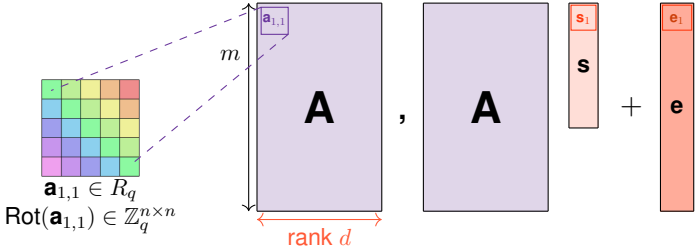
Using matrices, it gives the following block:

$$\begin{bmatrix} a_0 & -a_{n-1} & \cdots & -a_2 & -a_1 \\ a_1 & a_0 & \cdots & -a_3 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-2} & a_{n-3} & \cdots & a_0 & -a_{n-1} \\ a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ \vdots \\ s_{n-2} \\ s_{n-1} \end{bmatrix}$$

# Module LWE

Let $K$ be a number field of degree $n$ with $R$ its ring of integers.
Think of $K$ as $\mathbb{Q}[x]/(x^n+1)$ and of $R$ as $\mathbb{Z}[x]/(x^n+1)$ for $n = 2^k$.

Replace $\mathbb{Z}$ by $R$, and $\mathbb{Z}_q$ by $R_q = R/qR$.




- $\mathbf{A} \leftarrow U(R_q^{m\times d})$,
- $\mathbf{s} \leftarrow U(R_q^d)$,
- $e \in R^m$ small compared to $q$.

Special case $d = 1$ is Ring-LWE

# Module SIS and LWE

$R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ and $R_q = R/qR$.

Let $\alpha > 0$ and $\mathbf{s} \in (R_q)^d$, the distribution $A^{(M)}_{\mathbf{s}, D_{R,\alpha q}}$ is:

- $\mathbf{a} \in (R_q)^d$ uniform,
- $e$ sampled from $D_{R,\alpha q}$,

Outputs: $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + e)$.

Module-LWE$_{q,\nu_\alpha}$
Let $\mathbf{s} \in (R_q)^d$ uniform, distinguish between an arbitrary number of samples from $A^{(M)}_{\mathbf{s}, D_{R,\alpha q}}$ or the same number from $U((R_q)^d \times R_q)$.

# Ideals and modules

$R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ and $R_q = R/qR$.

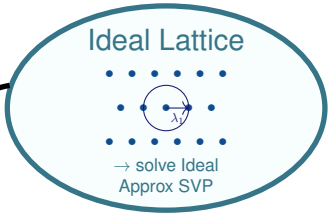
- An ideal $I$ of $R$ is an additive subgroup of $R$ closed under multiplication by every elements of $R$.
- As $R$ is isomorph to $\mathbb{Z}^n$, any ideal $I \in R$ defines an integer lattice $\Lambda(\mathbf{B})$ where $\mathbf{B} = \{g \bmod x^n + 1 : g \in I\}$.

- A subset $M \subseteq K^d$ is an $R$-module if it is closed under addition and multiplication by elements of $R$.
- A finite-type $R$-module: $M \subseteq R^d : \sum_{i=1}^{D} R \cdot \mathbf{b}_i$, $(\mathbf{b}_i) \in R^d$,
- $M = \sum_{i=1}^{d} I_i \cdot \mathbf{b}_i$ where $I_i$ are ideals of $R$ and $(I_i, \mathbf{b}_i)$ is a pseudo-basis of $M$.
- As ideals, any module defines an integer module lattice.

# Hardness of Ring Learning With Errors problem

Worst-case to average-case reduction

- **Stehlé, Steinfeld, Tanaka and Xagawa 2009** - search
- **Lyubashevsky, Peikert, Regev 2010** - decisional reduction both quantum, $q$ poly




Ring Learning With Errors

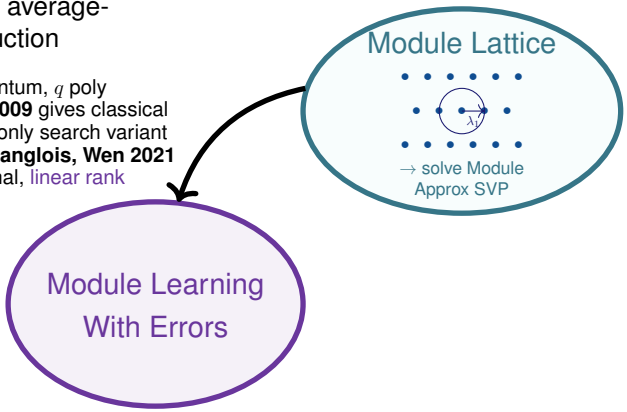
Self reductions

- **Applebaum, Cash, Peikert, Sahai 2009** - same error and secret

# Hardness of Module Learning With Errors problem

Worst-case to average-case reduction

- **Langlois Stehlé 2015** - quantum, $q$ poly
- Folklore: adapting **Peikert 2009** gives classical reduction but $q$ exp and only search variant
- **Boudgoust, Jeudy, Roux-Langlois, Wen 2021** classical, $q$ poly, decisional, linear rank




Self reductions

- **Applebaum, Cash, Peikert, Sahai 2009** - same error and secret
- **Boudgoust, Jeudy, Roux-Langlois, Wen 2022**: short error and secret distributions

# Module or Rings?

- Choice of parameters
  - Example of Ring $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$
  - Constraints on parameters $n = 2^k$, $q = 1 \bmod 2n$ ...
- An example of parameter set:
  - $n = 512 \Rightarrow$ 60 bits of security,
  - $n = 1024 \Rightarrow$ 140 bits of security,
  - $(n = 256, d = 3)$ gives $nd = 768$ which is "in between".
- Module LWE allows more flexibility.

# NIST competition

From 2017 to 2024, NIST competition to develop new standards on post-quantum cryptography

2022 first results: **3 over 4 new standards** are lattice-based

- Kyber - encryption scheme based on Module-LWE,
- Dilithium - signature scheme based on Module SIS and LWE,
- Falcon - signature scheme based on NTRU and Ring-SIS.

# Encryption scheme based on Ring-LWE

[Lyubashevsky, Peikert, Regev 2011]

KeyGen : The secret key is a small $s \in R$
The public key is $(a, b) = (a, b = a \cdot s + e) \in R_q^2$,
with $a \leftarrow U(R_q)$ and a small $e \in R$.

Enc : Given $m \in \{0,1\}^n$, a message is a polynomial in $R$ with coordinates in $\{0,1\}$. Sample small $r, e_1, e_2$ in $R$ and output

$$(a \cdot r + e_1, b \cdot r + e_2 + \lfloor q/2 \rfloor \cdot m) \in R_q \times R_q.$$

Dec : Given $(u, v) \in R_q \times R_q$, compute

$$v - u \cdot s = (r \cdot e - s \cdot e_1 + e_2) + b\lfloor q/2 \rfloor \cdot m$$

For each coordinate of $m$, the plaintext is $0$ if the result is closer from $0$ than $\lfloor q/2 \rfloor$, and $1$ otherwise.

# Kyber

[Avanzi, Bos, Ducas, Kiltz, Lepoint, Lyubashevsky, Schanck, Schwabe, Seiler, Stehle]

- Kyber relies on Module-LWE,
- Uses $R_q = \mathbb{Z}_q[x]/\langle x^{256} + 1\rangle$ with $q = 7681$.

- The small elements follow a binomial distribution $B_\eta$:
  For some positive integer $\eta$, sample $\{(ai, bi)\}_{i=1}^{\eta} \leftarrow (\{0,1\}^2)^\eta$ and output $\sum_{i=1}^{\eta}(a_i - b_i)$.

- The uniform public key is generated given a $seed$ and a function PARSE,
- Multiplication operations uses NTT - Number Theoretic Transform - which is a variant of the FFT in rings,
- Size of ciphertext is compressed by keeping only high order bits.

# Performances

Current timings (ECDH)
Public key around 32 bytes
Efficiency comparable in terms of cycles.

Kyber-512

| Sizes (in bytes) | | Haswell cycles (ref) | | Haswell cycles (avx2) | |
|---|---|---|---|---|---|
| sk: | 1632 | gen: | 122684 | gen: | 33856 |
| pk: | 800 | enc: | 154524 | enc: | 45200 |
| ct: | 768 | dec: | 187960 | dec: | 34572 |

Kyber-768

| Sizes (in bytes) | | Haswell cycles (ref) | | Haswell cycles (avx2) | |
|---|---|---|---|---|---|
| sk: | 2400 | gen: | 199408 | gen: | 52732 |
| pk: | 1184 | enc: | 235260 | enc: | 67624 |
| ct: | 1088 | dec: | 274900 | dec: | 53156 |

Kyber-1024

| Sizes (in bytes) | | Haswell cycles (ref) | | Haswell cycles (avx2) | |
|---|---|---|---|---|---|
| sk: | 3168 | gen: | 307148 | gen: | 73544 |
| pk: | 1568 | enc: | 346648 | enc: | 97324 |
| ct: | 1568 | dec: | 396584 | dec: | 79128 |

# Choice of parameters

- Parameters used by Kyber:
  - $n = 256$ and $d = 2, 3, 4$ giving three levels of security: $512, 768, 1024$,
  - $q = 7681$

# Choice of parameters

- Parameters used by Kyber:
  - $n = 256$ and $d = 2, 3, 4$ giving three levels of security: $512, 768, 1024$,
  - $q = 7681$

- How do they choose the parameters?
  - By considering the LWE instance with dimension $nd$,
  - and the ”lattice estimator” [Albrecht, Player, Scott 2015],

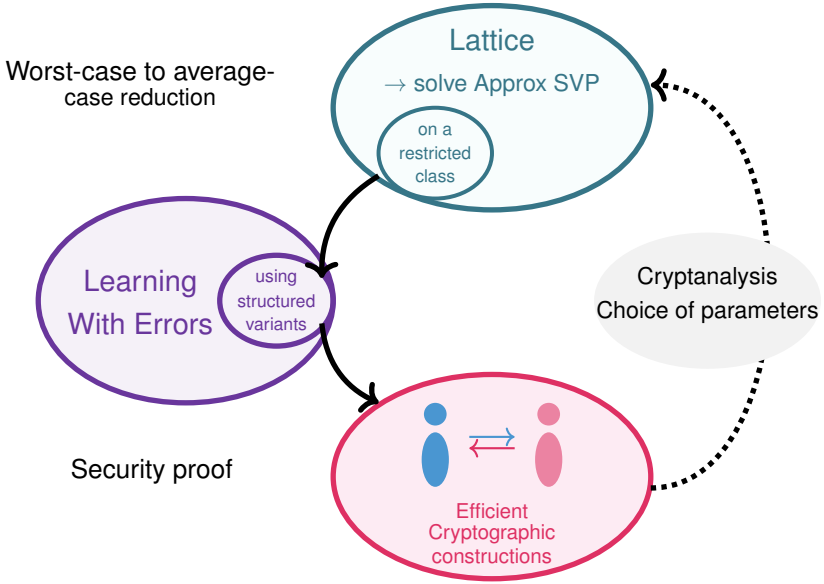
# Choice of parameters

- Parameters used by Kyber:
  - $n = 256$ and $d = 2, 3, 4$ giving three levels of security: $512, 768, 1024$,
  - $q = 7681$

- How do they choose the parameters?
  - By considering the LWE instance with dimension $nd$,
  - and the ”lattice estimator” [Albrecht, Player, Scott 2015],

- There is no consideration of the structure!
  - Why?
  - Because we don’t know how...

# Using LWE to build constructions in practice



# Conclusion

- Lattice-based cryptography allows to build efficient constructions such as encryption or signature schemes with a security based on the hardness of difficult algorithmic problems on lattices.

- Three schemes (Kyber, Dilithium and Falcon) will be standardise by the NIST, together with a hash-based signature.
  Two of them are based on Module-LWE.

- Approx Ideal SVP seems to be the easier problem to try to solve $\rightarrow$ the results of recent attacks does not impact the security of lattice-based constructions.