# Learning With Errors Problem
# Lattice Based Cryptography

Prof. dr. ir. Frederik Vercauteren

COSIC, KU Leuven

Computer Algebra for Cryptography (B-KUL-H0E74A)

2022-2023

Learning With Errors (LWE) problem

Algorithms for LWE problem

Lattice based cryptography

# Post-quantum public key cryptography

- Currently only two types PK are popular (see e.g. TLS 1.3 algorithms)
- Factoring based: mainly RSA
- Discrete logarithm based: DSA, ECDSA

# Post-quantum public key cryptography

- Currently only two types PK are popular (see e.g. TLS 1.3 algorithms)
- Factoring based: mainly RSA
- Discrete logarithm based: DSA, ECDSA
- **Shor (1994)**: quantum algorithm for factoring in time $O(\log^3 N)$, also computes discrete logarithms in polynomial time
- Initially: considered purely theoretical result. Now: threat taken seriously.
  - Even if Shor's algorithm is never implemented, the risk that one day it *could* is enough reason to change system (e.g. for long-term secrets)
  - History learns: long time between proposal and deployment

# Post-quantum public key cryptography

- Currently only two types PK are popular (see e.g. TLS 1.3 algorithms)
- Factoring based: mainly RSA
- Discrete logarithm based: DSA, ECDSA
- **Shor (1994)**: quantum algorithm for factoring in time $O(\log^3 N)$, also computes discrete logarithms in polynomial time
- Initially: considered purely theoretical result. Now: threat taken seriously.
  - Even if Shor's algorithm is never implemented, the risk that one day it *could* is enough reason to change system (e.g. for long-term secrets)
  - History learns: long time between proposal and deployment
- Need for new constructions for the post-quantum era (NIST):
  - Lattice based
  - Multivariate polynomial based
  - Code based
  - Hash based
  - Isogeny based

# Linear algebra over $\mathbb{Z}_q$

- Let $q$ be a prime and $\mathbb{Z}_q \simeq \mathbb{Z}/q\mathbb{Z}$ the field with $q$ elements
- System of $m$ linear equations in $n$ unknowns ($m \geq n$)

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \\ \vdots \\ c_m \end{pmatrix}$$

- Given matrix $A$ and vector $C$, Gaussian elimination finds $s_i$

# Distorting right hand side

- Instead of exact vector $C$, only given vector $B$ with

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \\ \vdots \\ c_m \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \\ \vdots \\ e_m \end{pmatrix}$$

- Error terms $e_i$ are small wrt. $q$ (in interval $[-q/2, q/2]$)
- Suddenly becomes very hard (not so over $\mathbf{Z}$, e.g. by least-squares method)
- Compare with disequations project:
  - disequations: every equation is *in*correct,
  - here: every equation is *almost* correct.

## Distorting right hand side

- Instead of exact vector $C$, only given vector $B$ with

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \\ \vdots \\ e_m \end{pmatrix}$$

- Error terms $e_i$ are small wrt. $q$ (in interval $[-q/2, q/2]$)
- Suddenly becomes very hard (not so over $\mathbf{Z}$, e.g. by least-squares method)
- Compare with disequations project:
  - disequations: every equation is *in*correct,
  - here: every equation is *almost* correct.

# Learning With Errors (LWE) problem: search

Regev (2005): *On lattices, learning with errors, random linear codes, and cryptography*

- Secret vector $\mathbf{s} \in \mathbb{Z}_q^n$ for some fixed $n$ and $q$
- An oracle generates random $\mathbf{a} \in \mathbb{Z}_q^n$ and a small error $e \leftarrow \chi$
- The oracle outputs $\mathbf{a}, b := \langle \mathbf{a}, \mathbf{s} \rangle + e \bmod q$ (linear almost-equation)
- Process is repeated many times for fresh $\mathbf{a}$ and $e$ (unlimited access to samples)

# Learning With Errors (LWE) problem: search

Regev (2005): *On lattices, learning with errors, random linear codes, and cryptography*

- Secret vector $\mathbf{s} \in \mathbb{Z}_q^n$ for some fixed $n$ and $q$
- An oracle generates random $\mathbf{a} \in \mathbb{Z}_q^n$ and a small error $e \leftarrow \chi$
- The oracle outputs $\mathbf{a}, b := \langle \mathbf{a}, \mathbf{s} \rangle + e \bmod q$ (linear almost-equation)
- Process is repeated many times for fresh $\mathbf{a}$ and $e$ (unlimited access to samples)

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \\ \vdots \\ e_m \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \\ \vdots \\ b_m \end{pmatrix}$$

# Learning With Errors (LWE) problem: example

- Secret vector $\mathbf{s} = [s_1, s_2, s_3, s_4] \in \mathbb{Z}_{13}^4$
- Given noisy inner products $\langle \mathbf{a}_i, \mathbf{s} \rangle$ of $\mathbf{s}$ with random vectors $\mathbf{a}_i$, try to recover $\mathbf{s}$

# Learning With Errors (LWE) problem: example

- Secret vector $\mathbf{s} = [s_1, s_2, s_3, s_4] \in \mathbb{Z}_{13}^4$
- Given noisy inner products $\langle \mathbf{a}_i, \mathbf{s} \rangle$ of $\mathbf{s}$ with random vectors $\mathbf{a}_i$, try to recover $\mathbf{s}$
- Each equation is correct up to small error $\{-1, 0, 1\}$

$$
\begin{aligned}
4 \cdot s_1 + 9 \cdot s_2 + 11 \cdot s_3 + 3 \cdot s_4 &\approx 9 \\
3 \cdot s_1 + 7 \cdot s_2 + 9 \cdot s_3 + 5 \cdot s_4 &\approx 5 \\
6 \cdot s_1 + 8 \cdot s_2 + 10 \cdot s_3 + 12 \cdot s_4 &\approx 3 \\
9 \cdot s_1 + 5 \cdot s_2 + 1 \cdot s_3 + 12 \cdot s_4 &\approx 3 \\
3 \cdot s_1 + 5 \cdot s_2 + 3 \cdot s_3 + 5 \cdot s_4 &\approx 10 \\
11 \cdot s_1 + 1 \cdot s_2 + 1 \cdot s_3 + 11 \cdot s_4 &\approx 5
\end{aligned}
$$

# Learning With Errors (LWE) problem: example

- Secret vector $\mathbf{s} = [s_1, s_2, s_3, s_4] \in \mathbb{Z}_{13}^4$
- Given noisy inner products $\langle \mathbf{a}_i, \mathbf{s} \rangle$ of $\mathbf{s}$ with random vectors $\mathbf{a}_i$, try to recover $\mathbf{s}$
- Each equation is correct up to small error $\{-1, 0, 1\}$

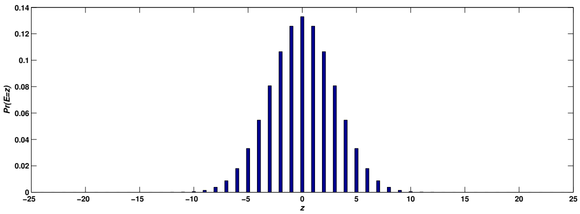
$$
\begin{aligned}
4 \cdot s_1 + 9 \cdot s_2 + 11 \cdot s_3 + 3 \cdot s_4 &\approx 9 \\
3 \cdot s_1 + 7 \cdot s_2 + 9 \cdot s_3 + 5 \cdot s_4 &\approx 5 \\
6 \cdot s_1 + 8 \cdot s_2 + 10 \cdot s_3 + 12 \cdot s_4 &\approx 3 \\
9 \cdot s_1 + 5 \cdot s_2 + 1 \cdot s_3 + 12 \cdot s_4 &\approx 3 \\
3 \cdot s_1 + 5 \cdot s_2 + 3 \cdot s_3 + 5 \cdot s_4 &\approx 10 \\
11 \cdot s_1 + 1 \cdot s_2 + 1 \cdot s_3 + 11 \cdot s_4 &\approx 5
\end{aligned}
$$

- Solution is [8, 3, 9, 2]
- **Exercise:** use MAGMA to find three other solutions
- If $m$ sufficiently large wrt to error rate $\approx \frac{\text{error size}}{q}$ then expect unique solution

# Discrete Gaussian distribution

- Theory: error distribution $\chi$ is discrete Gaussian distribution $\chi_s$ on $\mathbb{Z}$
  - Practice: error distribution is binomial distribution
- Width $s = \alpha q$ with error rate $\alpha < 1$
- Definition = discretization of continuous Gaussian distribution: for $z \in \mathbb{Z}$

$$\chi_s(z) = \frac{1}{C} \exp\left(\frac{-\pi z^2}{s^2}\right) \quad \text{with } C = \sum_{z \in \mathbf{Z}} \exp\left(\frac{-\pi z^2}{s^2}\right)$$




- Note: $\mu = 0$, $\sigma = s/\sqrt{2\pi}$

# Learning With Errors (LWE) problem: decision

Distinguish between two distributions:

| LWE distribution | Uniform distribution |
|---|---|
| Fixed $\mathbf{s} \in \mathbb{Z}_q^n$ | |
| $\mathbf{a}_i$ uniform random in $\mathbb{Z}_q^n$ | $\mathbf{a}_i$ uniform random in $\mathbb{Z}_q^n$ |
| $e_i$ small random error from $\chi$ | $b_i$ uniform random in $\mathbb{Z}_q$ |
| $(\mathbf{a}_1, b_1 := \langle \mathbf{a}_1, \mathbf{s} \rangle + e_1 \bmod q)$ | $(\mathbf{a}_1, b_1)$ |
| $(\mathbf{a}_2, b_2 := \langle \mathbf{a}_2, \mathbf{s} \rangle + e_2 \bmod q)$ | $(\mathbf{a}_2, b_2)$ |
| $\vdots$ | $\vdots$ |
| $(\mathbf{a}_m, b_m := \langle \mathbf{a}_m, \mathbf{s} \rangle + e_m \bmod q)$ | $(\mathbf{a}_m, b_m)$ |

- Hardness basically amounts to saying that $b_i$ look completely random

Learning With Errors (LWE) problem

Algorithms for LWE problem

Lattice based cryptography

# Naive algoritms

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & & & \end{pmatrix} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \end{pmatrix}$$

- Trial-and-error
  - Easy to test candidate-solution $\mathbf{s} \in \mathbb{Z}_q^n$: check that $b_i - \langle \mathbf{a}_i, \mathbf{s} \rangle$ is small for all $i$
  - $O(q^n)$ candidates
- Gaussian elimination?
  - Eliminate $a_{2,1}$ by computing $A[2] - a_{1,1}^{-1} a_{2,1} A[1]$
  - Element $a_{1,1}^{-1} a_{2,1}$ is typically large so blows up error $e_1$

# Naive algoritms

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & & & \end{pmatrix} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \end{pmatrix}$$

- Trial-and-error
  - Easy to test candidate-solution $\mathbf{s} \in \mathbb{Z}_q^n$: check that $b_i - \langle \mathbf{a}_i, \mathbf{s} \rangle$ is small for all $i$
  - $O(q^n)$ candidates
- Gaussian elimination?
  - Eliminate $a_{2,1}$ by computing $A[2] - a_{1,1}^{-1} a_{2,1} A[1]$
  - Element $a_{1,1}^{-1} a_{2,1}$ is typically large so blows up error $e_1$
  - Only combine equations with equal $a_{j,1}$ and $a_{k,1}$
  - Blum, Kalai, Wasserman '03: combine equations with equal blocks of coefficients
    - runs in time $2^{O(n)}$, best known algorithm but requires many samples

# Eliminating errors via lattices

- Given $\mathbf{b} \in \mathbb{Z}_q^{m\times 1}$ and $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ with $\mathbf{b} = \mathbf{A}\cdot\mathbf{s} + \mathbf{e}$
- Errors are small when reduced in the interval $[-q/2, q/2]$
- $\rightsquigarrow$ global problem with natural notion of smallness

# Eliminating errors via lattices

- Given $\mathbf{b} \in \mathbb{Z}_q^{m \times 1}$ and $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ with $\mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e}$
- Errors are small when reduced in the interval $[-q/2, q/2]$
- $\rightsquigarrow$ global problem with natural notion of smallness
- Consider the lattice in $\mathbb{Z}^m$

$$\mathcal{L}(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m \mid \mathbf{z} = \mathbf{A} \cdot \mathbf{x} \bmod q \text{ and } \mathbf{x} \in \mathbb{Z}_q^n\}$$

- Note that if $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{L}(\mathbf{A})$ we have $\mathbf{z}_1 - \mathbf{z}_2 \in \mathcal{L}(\mathbf{A})$
- **Exercise:** if $\mathbf{A}$ has rank $n$ then $\mathrm{vol}(\mathcal{L}(\mathbf{A})) = q^{m-n}$
- If $\mathbf{e} \neq 0$, then $\mathbf{b} \notin \mathcal{L}(\mathbf{A})$ but still quite close to it

# Eliminating errors via lattices

- Given $\mathbf{b} \in \mathbb{Z}_q^{m\times 1}$ and $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ with $\mathbf{b} = \mathbf{A}\cdot\mathbf{s} + \mathbf{e}$
- Errors are small when reduced in the interval $[-q/2, q/2]$
- $\rightsquigarrow$ global problem with natural notion of smallness
- Consider the lattice in $\mathbb{Z}^m$

$$\mathcal{L}(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m \mid \mathbf{z} = \mathbf{A}\cdot\mathbf{x} \bmod q \text{ and } \mathbf{x} \in \mathbb{Z}_q^n\}$$

- Note that if $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{L}(\mathbf{A})$ we have $\mathbf{z}_1 - \mathbf{z}_2 \in \mathcal{L}(\mathbf{A})$
- **Exercise:** if $\mathbf{A}$ has rank $n$ then $\mathrm{vol}(\mathcal{L}(\mathbf{A})) = q^{m-n}$
- If $\mathbf{e} \neq 0$, then $\mathbf{b} \notin \mathcal{L}(\mathbf{A})$ but still quite close to it
- **Bounded Distance Decoding** ($\mathrm{BDD}_d$): Given target vector with promise that it lies at distance $\leq d$ of lattice $\mathcal{L}$, find closest lattice vector = special case of CVP
- Note that the vector $\mathbf{b}$ is at distance $||\mathbf{e}||$ of $\mathcal{L}(\mathbf{A})$

# Eliminating errors via multivariate equations

- Arora, Ge (2011): *New algorithms for learning in the presence of errors*
- Given LWE samples $(\mathbf{a}_i, b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i)$, express algebraically that the errors $e_i$ are small
- Assuming errors are in $[-B, B]$, they will be zeros of

$$F(x) = \prod_{i \in [-B, B]} (x - i)$$

- Replacing secret vector $\mathbf{s}$ by unknowns $(x_1, \ldots, x_n)$ we can write

$$e_i = b_i - \langle \mathbf{a}_i, (x_1, \ldots, x_n) \rangle$$

- Obtain system of non-linear multivariate equations $F(e_i) = 0$
- For $\alpha q \sim n^{\epsilon}$, obtain complexity of $2^{\tilde{\mathcal{O}}(n^{2\epsilon})}$
- Useful if lot of samples and errors very small

# Properties of the LWE Problems

- **Theorem** (Regev '05): for $q$ prime with $q \leq n^{\mathcal{O}(1)}$, LWE is as hard as worst-case lattice problems ($\gamma$-SVP in dimension $n$) with $\gamma \sim n/\alpha$
  - Width of Gaussian should be large enough $\alpha q > 2\sqrt{n}$
  - Big theoretical result, main selling point of LWE (but in practice: smaller $\alpha q$)

# Properties of the LWE Problems

- **Theorem** (Regev '05): for $q$ prime with $q \leq n^{\mathcal{O}(1)}$, LWE is as hard as worst-case lattice problems ($\gamma$-SVP in dimension $n$) with $\gamma \sim n/\alpha$
  - Width of Gaussian should be large enough $\alpha q > 2\sqrt{n}$
  - Big theoretical result, main selling point of LWE (but in practice: smaller $\alpha q$)
- Given LWE problem with secret $\mathbf{s}$, can easily create LWE problem for secret $\mathbf{s} + \mathbf{t}$
  - Replace $b_i$ with $b_i + \langle \mathbf{a}_i, \mathbf{t} \rangle$
  - Random self-reduction

# Properties of the LWE Problems

- **Theorem** (Regev '05): for $q$ prime with $q \leq n^{\mathcal{O}(1)}$, LWE is as hard as worst-case lattice problems ($\gamma$-SVP in dimension $n$) with $\gamma \sim n/\alpha$
  - Width of Gaussian should be large enough $\alpha q > 2\sqrt{n}$
  - Big theoretical result, main selling point of LWE (but in practice: smaller $\alpha q$)
- Given LWE problem with secret $\mathbf{s}$, can easily create LWE problem for secret $\mathbf{s} + \mathbf{t}$
  - Replace $b_i$ with $b_i + \langle \mathbf{a}_i, \mathbf{t} \rangle$
  - Random self-reduction
- Search and decision problems are equivalent (easy for $q$ prime $O(\text{poly}(n))$)

# Search LWE $\leq_P$ Decision LWE

- Given an oracle that solves Decision LWE, we will solve the Search LWE
- Idea: use Decision oracle to deduce coefficients of $\mathbf{s}$ one at a time

# Search LWE $\leq_P$ Decision LWE

- Given an oracle that solves Decision LWE, we will solve the Search LWE
- Idea: use Decision oracle to deduce coefficients of $\mathbf{s}$ one at a time
- Make guess $g$ for the first coefficient of $\mathbf{s}$
- Change each sample $(\mathbf{a}, b)$ in $(\mathbf{a} + (r, 0, \ldots, 0), b + g \cdot r)$

# Search LWE $\leq_P$ Decision LWE

- Given an oracle that solves Decision LWE, we will solve the Search LWE
- Idea: use Decision oracle to deduce coefficients of $\mathbf{s}$ one at a time
- Make guess $g$ for the first coefficient of $\mathbf{s}$
- Change each sample $(\mathbf{a}, b)$ in $(\mathbf{a} + (r, 0, \ldots, 0), b + g \cdot r)$
- Submit new LWE instance to Decision oracle
  - If guess $g$ is correct, then new instance has LWE distribution
  - If guess $g$ is incorrect, then new instance has uniform distribution
- Repeat for other coefficients of $\mathbf{s}$

# Variants of LWE

- The secret $\mathbf{s}$ can be taken from the error distribution
- The secret $\mathbf{s}$ can be taken binary with sufficient entropy
- The noise vector $\mathbf{e}$ can be taken binomial, or uniform in some interval
- Learning With Rounding (LWR): the noise is computed deterministically as

$$(\mathbf{a} \in \mathbb{Z}_q^n, \lfloor \frac{p}{q} \langle \mathbf{a}, \mathbf{s} \rangle \rceil \bmod p)$$

  with $p$ a smaller integer than $q$ (idea: divide $\mathbf{Z}_q$ into $p$ intervals and round to starting point of interval; suffices to encode index of interval, hence mod $p$)
- LWR used in Saber (see later)

Learning With Errors (LWE) problem

Algorithms for LWE problem

Lattice based cryptography

# Cryptographic Applications of LWE

- LWE is as hard as worst case lattice problems, that are believed to be hard even for quantum computers
- Concrete security estimates via Albrecht's LWE tool: `https://bitbucket.org/malb/lwe-estimator`
- Very versatile! LWE has been used as the basis for:
  - Public key encryption
  - Identity-based encryption
  - Oblivious transfer
  - Leakage resilient encryption
  - Homomorphic encryption
  - ...
- Main downside: inefficient both in space and time (see in a couple of slides)

# Public Key Encryption based on LWE

- **Private key**: secret vector $\mathbf{s} \in \mathbb{Z}_q^n$ chosen uniform random
- **Public key**: $m$ samples from LWE distribution with secret $\mathbf{s}$, given as $m \times n$ matrix $A$ and $m \times 1$ matrix $B$

# Public Key Encryption based on LWE

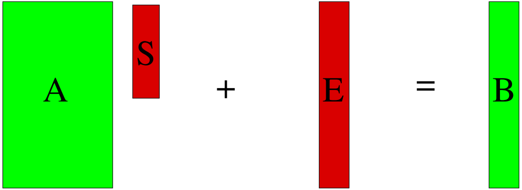
- **Private key**: secret vector $\mathbf{s} \in \mathbb{Z}_q^n$ chosen uniform random
- **Public key**: $m$ samples from LWE distribution with secret $\mathbf{s}$, given as $m \times n$ matrix $A$ and $m \times 1$ matrix $B$
- **Encryption**: for each bit $b$ of message do
  - choose random vector $\mathbf{r} \in \mathbb{Z}_q^m$ with small coefficients
  - ciphertext $= (\mathbf{c}, d) = (\mathbf{r}^t \cdot A, \mathbf{r}^t \cdot B + b \cdot \lfloor \frac{q}{2} \rfloor) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$

# Public Key Encryption based on LWE

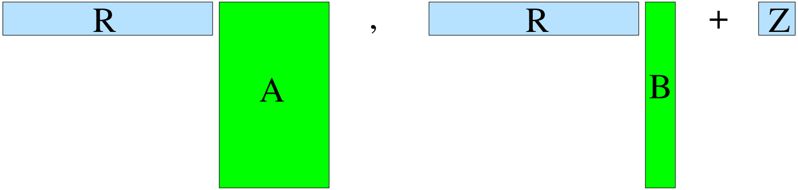
- **Private key**: secret vector $\mathbf{s} \in \mathbb{Z}_q^n$ chosen uniform random
- **Public key**: $m$ samples from LWE distribution with secret $\mathbf{s}$, given as $m \times n$ matrix $A$ and $m \times 1$ matrix $B$
- **Encryption**: for each bit $b$ of message do
  - choose random vector $\mathbf{r} \in \mathbb{Z}_q^m$ with small coefficients
  - ciphertext $= (\mathbf{c}, d) = (\mathbf{r}^t \cdot A, \mathbf{r}^t \cdot B + b \cdot \lfloor \frac{q}{2} \rfloor) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$
- **Decryption**: given ciphertext $(\mathbf{c}, d)$
  - if $d - \langle \mathbf{c}, \mathbf{s} \rangle$ closer to 0 than to $\lfloor \frac{q}{2} \rfloor$ modulo $q$, then message is 0 else it is 1

# Public Key Encryption based on LWE

Private/public key setup:




Encryption:



# Ring-LWE

- Main problem with LWE: requires $n$ elements in $\mathbb{Z}_q$ to generate only one extra random looking element in $\mathbb{Z}_q$

$$\mathbf{a}, b := \langle \mathbf{a}, \mathbf{s} \rangle + e$$

# Ring-LWE

- Main problem with LWE: requires $n$ elements in $\mathbb{Z}_q$ to generate only one extra random looking element in $\mathbb{Z}_q$

$$\mathbf{a}, b := \langle \mathbf{a}, \mathbf{s} \rangle + e$$

- Instead of inner product, try to use another type of product such that result is again in $\mathbb{Z}_q^n$ and not just $\mathbb{Z}_q$
- First idea: coordinate wise multiplication
    - Not secure since each coordinate is independent (one-dimensional LWE)
    - If $q \leq n^{\mathcal{O}(1)}$: easy search to find each coordinate of $\mathbf{s}$
    - If $q$ is very large: becomes related to Approximate GCD problem

# Ring-LWE

- Better idea: use multiplication in polynomial ring
- Consider $R := \mathbb{Z}[x]/(x^n + 1)$ with $n = 2^k$
- For an integer $q$, let $R_q = R/qR$
- Then can identify $\mathbb{Z}_q^n$ with $R_q$ by

$$[a_0, a_1, \ldots, a_{n-1}] \mapsto \sum_{i=0}^{n-1} a_i x^i$$

# Ring-LWE

- Better idea: use multiplication in polynomial ring
- Consider $R := \mathbb{Z}[x]/(x^n + 1)$ with $n = 2^k$
- For an integer $q$, let $R_q = R/qR$
- Then can identify $\mathbb{Z}_q^n$ with $R_q$ by

$$[a_0, a_1, \ldots, a_{n-1}] \mapsto \sum_{i=0}^{n-1} a_i x^i$$

- **Addition** is simply coordinate wise addition
- **Multiplication** is polynomial multiplication followed by reduction modulo $x^n + 1$

# Search Ring-LWE

- Example: $n = 4,\ q = 17$

$$a := 9x^3 + 8x^2 + 12x + 11 \quad s := x^3 + 12x^2 + 16x + 13$$

$$\Rightarrow a * s = 9x^3 + 11x^2 + 12x + 10$$

# Search Ring-LWE

- Example: $n = 4$, $q = 17$

$$a := 9x^3 + 8x^2 + 12x + 11 \quad s := x^3 + 12x^2 + 16x + 13$$

$$\Rightarrow a * s = 9x^3 + 11x^2 + 12x + 10$$

- Ring-LWE:
  - secret element $\mathbf{s} \in R_q$ (either small or random, equivalent)
  - elements $\mathbf{a}_i$ chosen randomly in $R_q$
  - coefficients noise polynomial $\mathbf{e}_i$ small independent normal variables
- **Search**: given many tuples $(\mathbf{a}_i, \mathbf{a}_i * \mathbf{s} + \mathbf{e}_i)$ recover $\mathbf{s}$
- Can be viewed as multiplication with structured $n \times n$ matrix
- Practice: $\mathbf{s}$ is taken to be small, so only one sample suffices

# Decision Ring-LWE

- **Decision**: given many tuples $(\mathbf{a}_i, \mathbf{b}_i) \in R_q^2$, decide whether there exists an $\mathbf{s} \in R_q$ and small $\mathbf{e}_i \in R_q$ such that
$$\mathbf{b}_i = \mathbf{a}_i * \mathbf{s} + \mathbf{e}_i$$

# Decision Ring-LWE

- **Decision**: given many tuples $(\mathbf{a}_i, \mathbf{b}_i) \in R_q^2$, decide whether there exists an $\mathbf{s} \in R_q$ and small $\mathbf{e}_i \in R_q$ such that

$$\mathbf{b}_i = \mathbf{a}_i * \mathbf{s} + \mathbf{e}_i$$

- If $q = 1 \bmod 2n$ and prime, then $x^n + 1$ has $n$ roots in $\mathbb{Z}_q$

$$\text{Search Ring-LWE} \leq_P \text{Decision Ring-LWE}$$

- Ring-LWE is as hard as worst case "structured lattice" (ideal lattice) problems
  - If $\mathbf{a}$ in lattice, then also $x * \mathbf{a}$
  - For $R$: if $(x_1, \ldots, x_n)$ in lattice, then also $(x_2, \ldots, x_n, -x_1)$

# Encryption based on RLWE

- Plaintext space is taken as $R_2$
- Let $\Delta = \lfloor q/2 \rfloor$
- Denote $[\cdot]_q$ reduction in $(-q/2, q/2]$
- $\chi$ error distribution on $R_q$

# Encryption based on RLWE

- Plaintext space is taken as $R_2$
- Let $\Delta = \lfloor q/2 \rfloor$
- Denote $[\cdot]_q$ reduction in $(-q/2, q/2]$
- $\chi$ error distribution on $R_q$
- **Secret key**: sample $\mathbf{s} \leftarrow \chi$

# Encryption based on RLWE

- Plaintext space is taken as $R_2$
- Let $\Delta = \lfloor q/2 \rfloor$
- Denote $[\cdot]_q$ reduction in $(-q/2, q/2]$
- $\chi$ error distribution on $R_q$
- **Secret key**: sample $\mathbf{s} \leftarrow \chi$
- **Public key**:
  - sample $\mathbf{a} \leftarrow R_q$, $\mathbf{e} \leftarrow \chi$ and output

$$\mathtt{pk} = \left( [(-\mathbf{a} \cdot \mathbf{s} + \mathbf{e})]_q \, , \mathbf{a} \right)$$

- Can interpret $\mathtt{pk}$ as degree 1 polynomial $\mathtt{pk}(X) = \mathtt{pk}[1]X + \mathtt{pk}[0]$ with

$$[\mathtt{pk}(\mathbf{s})]_q = \mathbf{e}$$

# Encryption based on RLWE

- **Encrypt** message $\mathbf{m} \in R_2$, let $\mathbf{p}_0 = \mathtt{pk}[0]$, $\mathbf{p}_1 = \mathtt{pk}[1]$
- Sample $\mathbf{u}, \mathbf{e}_1, \mathbf{e}_2 \leftarrow \chi$ and set

$$\mathtt{ct} = \left( [\mathbf{p}_0 \cdot \mathbf{u} + \mathbf{e}_1 + \Delta \cdot \mathbf{m}]_q , [\mathbf{p}_1 \cdot \mathbf{u} + \mathbf{e}_2]_q \right)$$

# Encryption based on RLWE

- **Encrypt** message $\mathbf{m} \in R_2$, let $\mathbf{p}_0 = \texttt{pk}[0]$, $\mathbf{p}_1 = \texttt{pk}[1]$
- Sample $\mathbf{u}, \mathbf{e}_1, \mathbf{e}_2 \leftarrow \chi$ and set

$$\texttt{ct} = \left( [\mathbf{p}_0 \cdot \mathbf{u} + \mathbf{e}_1 + \Delta \cdot \mathbf{m}]_q \, , [\mathbf{p}_1 \cdot \mathbf{u} + \mathbf{e}_2]_q \right)$$

- **Decrypt** ciphertext $\texttt{ct}$: set $\mathbf{c}_0 = \texttt{ct}[0]$, $\mathbf{c}_1 = \texttt{ct}[1]$ and compute

$$\left[ \left\lfloor \frac{[\mathbf{c}_0 + \mathbf{c}_1 \cdot \mathbf{s}]_q}{\Delta} \right\rceil \right]_2$$

# Decryption analysis

- Writing out definition

$$
\begin{aligned}
\mathbf{c}_0 + \mathbf{c}_1 \cdot \mathbf{s} &= \mathbf{p}_0 \cdot \mathbf{u} + \mathbf{e}_1 + \Delta \cdot \mathbf{m} + \mathbf{p}_1 \cdot \mathbf{u} \cdot \mathbf{s} + \mathbf{e}_2 \cdot \mathbf{s} \bmod q \\
&= \Delta \cdot \mathbf{m} + \mathbf{e} \cdot \mathbf{u} + \mathbf{e}_1 + \mathbf{e}_2 \cdot \mathbf{s} \bmod q
\end{aligned}
$$

# Decryption analysis

- Writing out definition

$$\begin{aligned}\mathbf{c}_0 + \mathbf{c}_1 \cdot \mathbf{s} &= \mathbf{p}_0 \cdot \mathbf{u} + \mathbf{e}_1 + \Delta \cdot \mathbf{m} + \mathbf{p}_1 \cdot \mathbf{u} \cdot \mathbf{s} + \mathbf{e}_2 \cdot \mathbf{s} \bmod q \\ &= \Delta \cdot \mathbf{m} + \mathbf{e} \cdot \mathbf{u} + \mathbf{e}_1 + \mathbf{e}_2 \cdot \mathbf{s} \bmod q\end{aligned}$$

- Error term $\mathbf{e} \cdot \mathbf{u} + \mathbf{e}_1 + \mathbf{e}_2 \cdot \mathbf{s}$ is small in $(-q/2, q/2]$
- As long as error term $< \Delta/2$ decryption works correctly

# Decryption analysis

- Writing out definition

$$\begin{aligned}\mathbf{c}_0 + \mathbf{c}_1 \cdot \mathbf{s} &= \mathbf{p}_0 \cdot \mathbf{u} + \mathbf{e}_1 + \Delta \cdot \mathbf{m} + \mathbf{p}_1 \cdot \mathbf{u} \cdot \mathbf{s} + \mathbf{e}_2 \cdot \mathbf{s} \bmod q \\ &= \Delta \cdot \mathbf{m} + \mathbf{e} \cdot \mathbf{u} + \mathbf{e}_1 + \mathbf{e}_2 \cdot \mathbf{s} \bmod q\end{aligned}$$

- Error term $\mathbf{e} \cdot \mathbf{u} + \mathbf{e}_1 + \mathbf{e}_2 \cdot \mathbf{s}$ is small in $(-q/2, q/2]$
- As long as error term $< \Delta/2$ decryption works correctly
- Valid ciphertext = deg 1 polynomial $\mathtt{ct}(X) = \mathtt{ct}[1]X + \mathtt{ct}[0]$ such that

$$[\mathtt{ct}(\mathbf{s})]_q = \Delta \cdot m + \mathbf{v}$$

with $|\mathbf{v}| < \Delta/2$

# Additively homomorphic property

- Let $\texttt{ct}_i$ for $i = 1, 2$ be two ciphertexts, with
$$[\texttt{ct}_i(\mathbf{s})]_q = \Delta \cdot \mathbf{m}_i + \mathbf{v}_i$$
then
$$[\texttt{ct}_1(\mathbf{s}) + \texttt{ct}_2(\mathbf{s})]_q = \Delta \cdot [\mathbf{m}_1 + \mathbf{m}_2]_2 + \mathbf{v}_1 + \mathbf{v}_2 + \epsilon,$$
where $\epsilon$ comes from reduction modulo 2 of $\mathbf{m}_1 + \mathbf{m}_2$
- Polynomial addition thus gives plaintext addition modulo 2
- Error grows additively in original errors
- Similar idea works for multiplication (requires relinearization: technical)
- Gentry's fantastic **bootstrapping** technique (2009): reduce the noise $\mathbf{v}$ without knowledge of $\mathbf{s}$ (very technical) $\rightsquigarrow$ Fully Homomorphic Encryption (FHE)

# NTRU

- Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman (1996): *NTRU: A Ring-Based Public Key Cryptosystem*
- Uses same polynomial ring as RLWE: $R_q = \mathbb{Z}_q[x]/(x^n + 1)$

# NTRU

- Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman (1996): *NTRU: A Ring-Based Public Key Cryptosystem*
- Uses same polynomial ring as RLWE: $R_q = \mathbb{Z}_q[x]/(x^n+1)$
- **Secret key**: $\mathbf{f}, \mathbf{g} \leftarrow \chi$ where $\chi$ samples elements with small coefficients in $R_q$ such that $\mathbf{f}$ is invertible in $R_q$
- **Public key**: $\mathbf{h} = \mathbf{g}/\mathbf{f} \in R_q$

# NTRU

- Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman (1996): *NTRU: A Ring-Based Public Key Cryptosystem*
- Uses same polynomial ring as RLWE: $R_q = \mathbb{Z}_q[x]/(x^n + 1)$
- **Secret key**: $\mathbf{f}, \mathbf{g} \leftarrow \chi$ where $\chi$ samples elements with small coefficients in $R_q$ such that $\mathbf{f}$ is invertible in $R_q$
- **Public key**: $\mathbf{h} = \mathbf{g}/\mathbf{f} \in R_q$
- **Plaintext space**: ring $R_p = \mathbb{Z}_p[x]/(x^n + 1)$ with $p$ much smaller than $q$, where each coefficient is taken in $[-p/2, p/2]$

# NTRU

- **Encryption**: to encrypt message $\mathbf{m} \in R_p$ under public key $\mathbf{h}$, generate polynomial $\mathbf{r} \leftarrow \chi'$ and compute
$$\mathbf{c} = p \cdot \mathbf{r} \cdot \mathbf{h} + \mathbf{m} \bmod q$$

# NTRU

- **Encryption**: to encrypt message $\mathbf{m} \in R_p$ under public key $\mathbf{h}$, generate polynomial $\mathbf{r} \leftarrow \chi'$ and compute
$$\mathbf{c} = p \cdot \mathbf{r} \cdot \mathbf{h} + \mathbf{m} \bmod q$$
- **Decryption**: given ciphertext $\mathbf{c}$ compute $\mathbf{c}' = \mathbf{f} \cdot \mathbf{c} \bmod q$ with all coefficients in $[-q/2, q/2]$
$$\mathbf{c}' = \mathbf{f} \cdot p \cdot \mathbf{r} \cdot \mathbf{h} + \mathbf{f} \cdot \mathbf{m} = p \cdot \mathbf{r} \cdot \mathbf{g} + \mathbf{f} \cdot \mathbf{m} \bmod q$$
- Note both terms $\mathbf{r} \cdot \mathbf{g}$ and $\mathbf{f} \cdot \mathbf{m}$ are small, so centered reduction really gives
$$\mathbf{c}' = p \cdot \mathbf{r} \cdot \mathbf{g} + \mathbf{f} \cdot \mathbf{m} \quad \text{(in } R\text{, i.e. no mod } q \text{ anymore!)}$$
- If $\mathbf{f} = 1 \bmod p$, then reduction modulo $p$ gives
$$\mathbf{c}' = \mathbf{m} \bmod p$$

# NTRU security

- NTRU problems:
  - Search: $\mathbf{h} \in R_q$, find $\mathbf{f}, \mathbf{g}$ with small coefficients and $\mathbf{h} = \mathbf{g}/\mathbf{f}$
  - Decision: distinguish $\mathbf{g}/\mathbf{f}$ from uniform random in $R_q$
  - 2021: Reduction due to A. Pellet-Mary and D. Stehlé

# NTRU security

- NTRU problems:
  - Search: $\mathbf{h} \in R_q$, find $\mathbf{f}, \mathbf{g}$ with small coefficients and $\mathbf{h} = \mathbf{g}/\mathbf{f}$
  - Decision: distinguish $\mathbf{g}/\mathbf{f}$ from uniform random in $R_q$
  - 2021: Reduction due to A. Pellet-Mary and D. Stehlé
- Link with lattices: the private key $\mathbf{g}, \mathbf{f}$ is small and satisfies

$$\mathbf{f} \cdot \mathbf{h} = \mathbf{g} \bmod q$$

- Vector $(\mathbf{f}, \mathbf{g})$ is very short vector in lattice

$$\{ (\mathbf{u}, \mathbf{v}) \in R^2 \mid \mathbf{uh} \equiv \mathbf{v} \bmod q \} \subset \mathbb{Z}^{2n}$$

  (inclusion via identification of $R$ with $\mathbb{Z}^n$)
- **Exercise:** show that this is the lattice

$$\begin{pmatrix} I_n & \mathrm{rot}^-(\mathbf{h}) \\ 0 & qI_n \end{pmatrix}$$

  where $i$-th row of $\mathrm{rot}^-(\mathbf{h})$ is simply $x^i\mathbf{h} \bmod q$

# NIST competition

- Standardization effort for post-quantum cryptographic schemes
- November 2017: 69 accepted submissions
- July 2020: 7 finalists and 8 alternates

| Type | Key encapsulation | Digital signature |
|---|---|---|
| Lattice | Kyber, NTRU, Saber | Dilithium, Falcon |
| | FrodoKEM, NTRU Prime | |
| Code | Classic McEliece, BIKE, HQC | – |
| Multivariate | – | Rainbow, GeMMS |
| Isogeny | SIKE | – |
| Hash | – | SPHINCS+ |
| ZK proofs | – | Picnic |

- Saber = Cosic submission
-
-

# NIST competition

- Standardization effort for post-quantum cryptographic schemes
- November 2017: 69 accepted submissions
- July 2020: 7 finalists and 8 alternates

| Type | Key encapsulation | Digital signature |
| --- | --- | --- |
| Lattice | Kyber, NTRU, Saber<br>FrodoKEM, NTRU Prime | Dilithium, Falcon |
| Code | Classic McEliece, BIKE, HQC | – |
| Multivariate | – | ~~Rainbow~~, GeMMS |
| Isogeny | ~~SIKE~~ | – |
| Hash | – | SPHINCS+ |
| ZK proofs | – | Picnic |

- Saber = Cosic submission
- July 22: 4 standards, 4 for further scrutiny (also in 2022: 2 complete breaks)
- New call for digital signatures (submission deadline June 2023)

# Further reading

- O. Regev. The Learning with Errors Problem:
  http://www.cims.nyu.edu/~regev/papers/lwesurvey.pdf
- S. Galbraith. Cryptosystems based on lattices:
  https://www.math.auckland.ac.nz/~sgal018/crypto-book/ch19a.pdf