The LWE problem
from lattices to cryptography

Damien Stehlé
ENS de Lyon

Šibenik, June 2015
What is a good problem, for a cryptographer?

- Almost all of its instances must be hard to solve.
  
  Attacks must be too expensive.

- Its instances must be easy to sample.
  
  The algorithms run by honest users should be efficient.

- The problem must be (algebraically) rich/expressive.
  
  So that interesting models of attacks can be handled, even for advanced cryptographic functionalities.
What is a good problem, for a cryptographer?

- Almost all of its instances must be **hard to solve**.
  
  Attacks must be too expensive.

- Its instances must be **easy to sample**.
  
  The algorithms run by honest users should be efficient.

- The problem must be (algebraically) **rich/expresive**.
  
  So that interesting models of attacks can be handled, even for advanced cryptographic functionalities.
What is a good problem, for a cryptographer?

- Almost all of its instances must be hard to solve.
  
  Attacks must be too expensive.

- Its instances must be easy to sample.
  
  The algorithms run by honest users should be efficient.

- The problem must be (algebraically) rich/expressive.
  
  So that interesting models of attacks can be handled, even for advanced cryptographic functionalities.
What is a good problem, for a cryptographer?

- Almost all of its instances must be **hard to solve**.
  Attacks must be too expensive.

- **Its instances must be easy to sample.**
  The algorithms run by honest users should be efficient.

- **The problem must be (algebraically) rich/expressive.**
  So that interesting models of attacks can be handled, even for advanced cryptographic functionalities.
The Learning With Errors problem

**Informal definition**

Solve a random system of $m$ noisy linear equations and $n$ unknowns modulo an integer $q$, with $m \gg n$.

- The best known algorithms are exponential in $n \log q$.
- Sampling an instance costs $O(mn \log q)$. Very often, $m = O(n \log q)$, so this is $O((n \log q)^2)$.
- Very rich/expressive: encryption [Re05], ID-based encr. [GePeVa08], fully homomorphic encr. [BrVa11], attribute-based encr. [GoVaWe13], etc.
The Learning With Errors problem

Informal definition
Solve a random system of $m$ noisy linear equations and $n$ unknowns modulo an integer $q$, with $m \gg n$.

- The best known algorithms are exponential in $n \log q$.
- Sampling an instance costs $O(mn \log q)$.
  Very often, $m = O(n \log q)$, so this is $O((n \log q)^2)$.
- Very rich/expressive:
  encryption [Re05], ID-based encr. [GePeVa08], fully homomorphic encr. [BrVa11], attribute-based encr. [GoVaWe13], etc.
The Learning With Errors problem

Informal definition

Solve a random system of $m$ noisy linear equations and $n$ unknowns modulo an integer $q$, with $m \gg n$.

- The best known algorithms are exponential in $n \log q$.
- Sampling an instance costs $O(mn \log q)$.
  Very often, $m = O(n \log q)$, so this is $O((n \log q)^2)$.
- Very rich/expressive:
  encryption [Re05], ID-based encr. [GePeVa08], fully homomorphic encr. [BrVa11], attribute-based encr. [GoVaWe13], etc.
Goals of this talk

- Introduce LWE.
- Show the relationship between LWE and lattices.
- Use LWE to design a public-key encryption scheme.
- Give some open problems.
Road-map

- Definition of the LWE problem
- Regev’s encryption scheme
- Lattice problems
- Hardness of LWE
- Equivalent problems
Road-map

- Definition of the LWE problem
- Regev’s encryption scheme
- Lattice problems
- Hardness of LWE
- Equivalent problems
Gaussian distributions

Continuous Gaussian of parameter $s$:

$$D_s(x) \sim \frac{1}{s} \exp \left( -\pi \frac{x^2}{s^2} \right) \quad \forall x \in \mathbb{R}$$
Gaussian distributions

Continuous Gaussian of parameter $s$:

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

Discrete Gaussian of support $\mathbb{Z}$ and parameter $s$:

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

Continuous Gaussian of parameter $s$:

\[
D_s(x) \sim \frac{1}{s} \exp \left( -\pi \frac{x^2}{s^2} \right)
\]
\[\forall x \in \mathbb{R}\]

Discrete Gaussian of support $\mathbb{Z}$ and parameter $s$:

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

- That’s not the rounding of a continuous Gaussian.
- One may efficiently sample from it.
- The usual tail bound holds.
The LWE problem \cite{Re05}

Let $n \geq 1$, $q \geq 2$ and $\alpha \in (0, 1)$. For all $s \in \mathbb{Z}_q^n$, we define the distribution $D_{n,q,\alpha}(s)$:

$$(a, \langle a, s \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q,$$

with $a \leftarrow U(\mathbb{Z}_q^n)$ and $e \leftarrow D_{\mathbb{Z},\alpha q}$.

**Search LWE**

For all $s$: Given arbitrarily many samples from $D_{n,q,\alpha}(s)$, find $s$.

(Information-theoretically, $\approx n \frac{\log q}{\log \frac{1}{\alpha}}$ samples uniquely determine $s$.)

**Decision LWE**

With non-negligible probability over $s \leftarrow U(\mathbb{Z}_q^n)$:

distinguish between the distributions $D_{n,q,\alpha}(s)$ and $U(\mathbb{Z}_q^{n+1})$.

(Non-negligible: $1/(n \log q)^c$ for some constant $c > 0$.)
The LWE problem [Re05]

Let $n \geq 1$, $q \geq 2$ and $\alpha \in (0, 1)$.
For all $s \in \mathbb{Z}_q^n$, we define the distribution $D_{n,q,\alpha}(s)$:

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

Search LWE

For all $s$: Given arbitrarily many samples from $D_{n,q,\alpha}(s)$, find $s$.

(Information-theoretically, $\approx n \frac{\log q}{\log 1/\alpha}$ samples uniquely determine $s$.)

Decision LWE

With non-negligible probability over $s \leftarrow U(\mathbb{Z}_q^n)$:
distinguish between the distributions $D_{n,q,\alpha}(s)$ and $U(\mathbb{Z}_q^{n+1})$.

(Non-negligible: $1/(n \log q)^c$ for some constant $c > 0$.)
Let $n \geq 1$, $q \geq 2$ and $\alpha \in (0, 1)$. For all $s \in \mathbb{Z}_q^n$, we define the distribution $D_{n,q,\alpha}(s)$:

$$(a, \langle a, s \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q,$$

with $a \leftarrow U(\mathbb{Z}_q^n)$ and $e \leftarrow D_{\mathbb{Z},\alpha q}$.

**Search LWE**

For all $s$: Given arbitrarily many samples from $D_{n,q,\alpha}(s)$, find $s$.

(Information-theoretically, $\approx n \frac{\log q}{\log \frac{1}{\alpha}}$ samples uniquely determine $s$.)

**Decision LWE**

With non-negligible probability over $s \leftarrow U(\mathbb{Z}_q^n)$: distinguish between the distributions $D_{n,q,\alpha}(s)$ and $U(\mathbb{Z}_q^{n+1})$.

(Non-negligible: $1/(n \log q)^c$ for some constant $c > 0$.)
Decision LWE

Let $n \geq 1$, $q \geq 2$ and $\alpha \in (0, 1)$.
For all $s \in \mathbb{Z}_q^n$, we define the distribution $D_{n,q,\alpha}(s)$:

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

We are given an oracle $\mathcal{O}$ that produces independent samples from always the same distribution, which is:

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

We have to tell which, with probability $\geq \frac{1}{2} + \frac{1}{(n \log q)^\Omega(1)}$. 

**Decision LWE**

With non-negligible probability over $s \leftarrow U(\mathbb{Z}_q^n)$:
distinguish between the distributions $D_{n,q,\alpha}(s)$ and $U(\mathbb{Z}_q^{n+1})$. 

Damien Stehlé
Search LWE $\equiv$ solving noisy linear systems

Find $s_1, s_2, s_3, s_4, s_5 \in \mathbb{Z}_{23}$ such that:

\[
\begin{align*}
    s_1 + 22s_2 + 17s_3 + 2s_4 + s_5 & \approx 16 \mod 23 \\
    3s_1 + 2s_2 + 11s_3 + 7s_4 + 8s_5 & \approx 17 \mod 23 \\
    15s_1 + 13s_2 + 10s_3 + s_4 + 22s_5 & \approx 3 \mod 23 \\
    17s_1 + 11s_2 + s_3 + 10s_4 + 3s_5 & \approx 8 \mod 23 \\
    2s_1 + s_2 + 13s_3 + 6s_4 + 2s_5 & \approx 9 \mod 23 \\
    4s_1 + 4s_2 + s_3 + 5s_4 + s_5 & \approx 18 \mod 23 \\
    11s_1 + 12s_2 + 5s_3 + s_4 + 9s_5 & \approx 7 \mod 23
\end{align*}
\]

We can even ask for arbitrarily many noisy equations.
Matrix version of LWE

\[
\begin{pmatrix}
\text{A} & \text{A} \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
m \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
n \\
\end{pmatrix}
\]

\[
\text{A} \leftarrow U(\mathbb{Z}_q^{m \times n}),
\]

\[
\text{s} \leftarrow U(\mathbb{Z}_q^n),
\]

\[
\text{e} \leftarrow D_{\mathbb{Z}_q^m, \alpha q}.
\]

**Discrete Gaussian error**

**Decision LWE:**

Determine whether \((\text{A}, \text{b})\) is of the form above, or uniform.
Some simple remarks

- If $\alpha \approx 0$, LWE is easy to solve.
- If $\alpha \approx 1$, LWE is trivially hard.
- Very often, we are interested in
  \[ \alpha \approx \frac{1}{n^c}, \quad q \approx n^{c'}, \quad \text{for some constants } c' > c > 0. \]
- Why a discrete Gaussian noise?
Why is LWE interesting for crypto?

- LWE is just noisy linear algebra: Easy to use, expressive.
- LWE seems to be a (very) hard problem.

Two particularly useful properties:
- Unlimited number of samples.
- Random self-reducibility over $s$.

If $q$ is prime and $\leq n^{O(1)}$, there are polynomial-time reductions between the Search and Decision versions of LWE [Re05].

(We may remove these assumptions, if we allow some polynomial blow-up on $\alpha$.)
Why is LWE interesting for crypto?

- LWE is just noisy linear algebra: Easy to use, expressive.
- LWE seems to be a (very) hard problem.

Two particularly useful properties:
- Unlimited number of samples.
- Random self-reducibility over $s$.

If $q$ is prime and $\leq n^{O(1)}$, there are polynomial-time reductions between the Search and Decision versions of LWE [Re05].

(We may remove these assumptions, if we allow some polynomial blow-up on $\alpha$.)
Road-map

- Definition of the LWE problem
- Regev’s encryption scheme
- Lattice problems
- Hardness of LWE
- Equivalent problems
A public-key encryption scheme over \( \{0, 1\} \times \mathcal{C} \) consists in three algorithms:

- **\textsc{KeyGen}**: Security parameter \( \mapsto (pk, sk) \).
- **\textsc{Enc}**: \( (pk, M) \mapsto C \in \mathcal{C} \).
- **\textsc{Dec}**: \( (sk, C) \mapsto M' \in \{0, 1\} \).

**Correctness**

With probability \( \approx 1 \), \( \forall M \in \{0, 1\} : \textsc{Dec}_{sk}(\textsc{Enc}_{pk}(M)) = M \).

**Security (IND-CPA)**

The distributions of \( (pk, \textsc{Enc}_{pk}(0)) \) and \( (pk, \textsc{Enc}_{pk}(1)) \) must be \textit{computationally indistinguishable}. 
Public-key encryption

A public-key encryption scheme over \( \{0, 1\} \times \mathcal{C} \) consists in three algorithms:

- **\texttt{KeyGen}**: Security parameter \( \mapsto (pk, sk) \).
- **\texttt{Enc}**: \((pk, M) \mapsto C \in \mathcal{C} \).
- **\texttt{Dec}**: \((sk, C) \mapsto M' \in \{0, 1\} \).

**Correctness**

With probability \( \approx 1 \), \( \forall M \in \{0, 1\} : \text{\texttt{Dec}}_{sk}(\text{\texttt{Enc}}_{pk}(M)) = M \).

**Security** (IND-CPA)

The distributions of \((pk, \text{\texttt{Enc}}_{pk}(0))\) and \((pk, \text{\texttt{Enc}}_{pk}(1))\) must be computationally indistinguishable.
Public-key encryption

A public-key encryption scheme over \( \{0, 1\} \times C \) consists in three algorithms:

- **KEYGEN**: Security parameter \( \mapsto (pk, sk) \).
- **ENC**: \( (pk, M) \mapsto C \in C \).
- **DEC**: \( (sk, C) \mapsto M' \in \{0, 1\} \).

**Correctness**

With probability \( \approx 1 \), \( \forall M \in \{0, 1\} : \text{DEC}_{sk}(\text{ENC}_{pk}(M)) = M \).

**Security (IND-CPA)**

The distributions of \( (pk, \text{ENC}_{pk}(0)) \) and \( (pk, \text{ENC}_{pk}(1)) \) must be **computationally indistinguishable**.
Parameters: \( n, m, q, \alpha \).

Keys: \( \text{sk} = s \) and \( \text{pk} = (A, b) \), with \( b = A \cdot s + e \).

\( \text{ENC}(M \in \{0, 1\}) \): Let \( r \sim U(\{0, 1\}^m) \),

\[
\begin{align*}
u^T &= A, \quad v = b + \left\lfloor \frac{q}{2} \right\rfloor \cdot M.
\end{align*}
\]

\( \text{DEC}(u, v) \): Compute \( v - u^T s \) (modulo \( q \)).

If it’s close to 0, output 0, else output 1.
Regev’s encryption scheme

- **Parameters**: \( n, m, q, \alpha \).
- **Keys**: \( \text{sk} = s \) and \( \text{pk} = (A, b) \), with \( b = As + e \).
- **ENC\((M \in \{0, 1\})\)**: Let \( r \leftarrow U(\{0, 1\}^m) \),
  \[
  u^T = \begin{bmatrix} r^T \\ \end{bmatrix}, \quad v = \begin{bmatrix} r^T \\ \end{bmatrix}A + \left\lfloor \frac{q}{2} \right\rfloor M.
  \]

- **DEC\((u, v)\)**: Compute \( v - u^T s \) (modulo \( q \)).
  \[
  \begin{bmatrix} r^T \\ \end{bmatrix}A + \left\lfloor \frac{q}{2} \right\rfloor M - \]
  \[
  = \text{small} + \left\lfloor \frac{q}{2} \right\rfloor M
  \]

If it’s close to 0, output 0, else output 1.
Decryption correctness

Correctness

Assume that \( \alpha \leq o\left(\frac{1}{\sqrt{m\log n}}\right) \).
Then, with probability \( \geq 1 - n^{-\omega(1)} \), it correctly decrypts.

We have

\[ v - u^T s = r^T e + \lfloor q/2 \rfloor M \mod q. \]

As \( e \sim D_{\mathbb{Z},\alpha q}^m \), we expect \( \langle r, e \rangle \) to behave like \( D_{\|r\|\alpha q} \).

As \( \|r\| \leq \sqrt{m} \), we have \( \|r\|\alpha q \leq o\left(\frac{q}{\sqrt{\log n}}\right) \), and

a sample from \( D_{\|r\|\alpha q} \) is \( < q/8 \) with probability \( \geq 1 - n^{-\omega(1)} \).
Decryption correctness

Assume that $\alpha \leq o\left(\frac{1}{\sqrt{m \log n}}\right)$.
Then, with probability $\geq 1 - n^{-\omega(1)}$, it correctly decrypts.

We have

$$v - u^T s = r^T e + \lfloor q/2 \rfloor M \mod q.$$ 

As $e \sim D_{\mathbb{Z},\alpha q}^m$, we expect $\langle r, e \rangle$ to behave like $D_{\|r\|\alpha q}$.

As $\|r\| \leq \sqrt{m}$, we have $\|r\|\alpha q \leq o\left(\frac{q}{\sqrt{\log n}}\right)$, and
a sample from $D_{\|r\|\alpha q}$ is $< q/8$ with probability $\geq 1 - n^{-\omega(1)}$.

$\Rightarrow$ We know $r^T e + \lfloor q/2 \rfloor M$ over the integers.
Decryption correctness

Correctness

Assume that $\alpha \leq o\left(\frac{1}{\sqrt{m\log n}}\right)$.
Then, with probability $\geq 1 - n^{-\omega(1)}$, it correctly decrypts.

We have

$$v - u^T s = r^T e + \lfloor q/2 \rfloor M \mod q.$$

As $e \sim D_{\mathbb{Z}, \alpha q}^m$, we expect $\langle r, e \rangle$ to behave like $D_{\|r\|\alpha q}$.

As $\|r\| \leq \sqrt{m}$, we have $\|r\|\alpha q \leq o\left(\frac{q}{\sqrt{\log n}}\right)$, and

a sample from $D_{\|r\|\alpha q}$ is $< q/8$ with probability $\geq 1 - n^{-\omega(1)}$.

$\Rightarrow$ We know $r^T e + \lfloor q/2 \rfloor M$ over the integers.
Decryption correctness

Correctness

Assume that $\alpha \leq o\left(\frac{1}{\sqrt{m \log n}}\right)$.
Then, with probability $\geq 1 - n^{-\omega(1)}$, it correctly decrypts.

We have

$$v - u^T s = r^T e + \lfloor q/2 \rfloor M \mod q.$$ 

As $e \sim D_{\mathbb{Z}, \alpha q}^m$, we expect $\langle r, e \rangle$ to behave like $D_{\|r\|\alpha q}$.

As $\|r\| \leq \sqrt{m}$, we have $\|r\|\alpha q \leq o\left(\frac{q}{\sqrt{\log n}}\right)$, and

a sample from $D_{\|r\|\alpha q}$ is $< q/8$ with probability $\geq 1 - n^{-\omega(1)}$.

$\Rightarrow$ We know $r^T e + \lfloor q/2 \rfloor M$ over the integers.
IND-CPA Security

Security

Assume that \( m = \Omega(n \log q) \). Then any (IND-CPA) attacker may be turned into an algorithm for LWE\(_{n,q,\alpha}\).

Fake security experiment

Challenger uses and gives to the attacker a uniform pair \((A, b)\) (instead of \(b = A \cdot s + e\)).

- If attacker behaves differently than in real security experiment, it can be used to solve LWE.
- In fake experiment, \((A, b, r^T A, r^T b)\) is \(\approx\) uniform, hence \(Enc(0)\) and \(Enc(1)\) follow \(\approx\) the same distribution.
IND-CPA Security

Assume that $m = \Omega(n \log q)$. Then any (IND-CPA) attacker may be turned into an algorithm for LWE$_{n,q,\alpha}$.

Fake security experiment

Challenger uses and gives to the attacker a uniform pair $(A, b)$ (instead of $b = A \cdot s + e$).

1. If attacker behaves differently than in real security experiment, it can be used to solve LWE.
2. In fake experiment, $(A, b, r^T A, r^T b)$ is $\approx$ uniform, hence $\text{ENC}(0)$ and $\text{ENC}(1)$ follow ($\approx$) the same distribution.
IND-CPA Security

Security
Assume that \( m = \Omega(n \log q) \). Then any (IND-CPA) attacker may be turned into an algorithm for LWE_{n,q,\alpha}.

Fake security experiment
Challenger uses and gives to the attacker a uniform pair \((A, b)\) (instead of \(b = A \cdot s + e\)).

1. If attacker behaves differently than in real security experiment, it can be used to solve LWE.
2. In fake experiment, \((A, b, r^T A, r^T b)\) is \(\approx\) uniform, hence \(\text{ENC}(0)\) and \(\text{ENC}(1)\) follow \(\approx\) the same distribution.
IND-CPA Security

**Security**

Assume that $m = \Omega(n \log q)$. Then any (IND-CPA) attacker may be turned into an algorithm for $\text{LWE}_{n,q,\alpha}$.

**Fake security experiment**

Challenger uses and gives to the attacker a uniform pair $(A, b)$ (instead of $b = A \cdot s + e$).

1. If attacker behaves differently than in real security experiment, it can be used to solve LWE.
2. In fake experiment, $(A, b, r^T A, r^T b)$ is $\approx$ uniform, hence $\text{ENC}(0)$ and $\text{ENC}(1)$ follow ($\approx$) the same distribution.
Setting the parameters: \( n, m, \alpha, q \)

- Correctness: \( \alpha \leq o\left(\frac{1}{\sqrt{m \log n}}\right) \)
- Reducing LWE to IND-CPA security: \( m \geq \Omega(n \log q) \)

1. Set \( \alpha \) as large as possible (\( \alpha \) impacts security)
2. Set \( m \) as small as possible (\( m \) impacts efficiency)
3. Set \( n \) and \( q \) so that LWE\(_{n,q,\alpha} \) is sufficiently hard to solve

Here: \( \alpha = \tilde{\Theta}(\sqrt{n}) \), \( m = \tilde{\Theta}(n) \) and \( q = \tilde{\Theta}(n) \).

This is not very practical... ciphertext expansion: \( \tilde{\Theta}(n) \).
Setting the parameters: $n, m, \alpha, q$

- Correctness: $\alpha \leq o\left(\frac{1}{\sqrt{m \log n}}\right)$
- Reducing LWE to IND-CPA security: $m \geq \Omega(n \log q)$

1. Set $\alpha$ as large as possible ($\alpha$ impacts security)
2. Set $m$ as small as possible ($m$ impacts efficiency)
3. Set $n$ and $q$ so that $\text{LWE}_{n,q,\alpha}$ is sufficiently hard to solve

Here: $\alpha = \widetilde{\Theta}(\sqrt{n})$, $m = \widetilde{\Theta}(n)$ and $q = \widetilde{\Theta}(n)$.

This is not very practical... ciphertext expansion: $\widetilde{\Theta}(n)$. 

Damien Stehlé
Setting the parameters: \( n, m, \alpha, q \)

- Correctness: \( \alpha \leq o\left(\frac{1}{\sqrt{m \log n}}\right) \)
- Reducing LWE to IND-CPA security: \( m \geq \Omega(n \log q) \)

1. Set \( \alpha \) as large as possible \((\alpha \text{ impacts security})\)
2. Set \( m \) as small as possible \((m \text{ impacts efficiency})\)
3. Set \( n \) and \( q \) so that LWE\(_{n,q,\alpha} \) is sufficiently hard to solve

Here: \( \alpha = \tilde{\Theta}(\sqrt{n}), \ m = \tilde{\Theta}(n) \) and \( q = \tilde{\Theta}(n) \).

This is not very practical... ciphertext expansion: \( \tilde{\Theta}(n) \).
Multi-bit Regev

- **Parameters:** $n, m, q, \alpha, \ell$.
- **Keys:** $sk = S \in \mathbb{Z}_q^{n \times \ell}$ and $pk = (A, B)$, with $B = AS + E$.
- **ENC($M \in \{0, 1\}^\ell$):** Let $r \leftarrow U(\{0, 1\}^m)$,
  
  \[
  u^T = \begin{bmatrix} A \\ \hline \end{bmatrix}, \quad v^T = \begin{bmatrix} B + [q/2] \cdot M^T \\ \hline \end{bmatrix}.
  \]

- **DEC($u, v$):** Compute $v^T - u^T S$ (modulo $q$).

### Asymptotic performance, for $\ell = n$

- Ciphertext expansion: $\tilde{\Theta}(1)$
- Processing time: $\tilde{\Theta}(n)$ per message bit
- Key size: $\tilde{\Theta}(n^2)$
Multi-bit Regev

- **Parameters:** $n, m, q, \alpha, \ell$.
- **Keys:** $sk = S \in \mathbb{Z}_q^{n \times \ell}$ and $pk = (A, B)$, with $B = AS + E$.
- **ENC($M \in \{0, 1\}^\ell$):** Let $r \leftarrow U(\{0, 1\}^m)$,

$$u^T = A^T, \quad v^T = B^T + \lfloor q/2 \rfloor \cdot M^T.$$

- **DEC($u, v$):** Compute $v^T - u^T S$ (modulo $q$).

**Asymptotic performance, for $\ell = n$**

- Ciphertext expansion: $\tilde{\Theta}(1)$
- Processing time: $\tilde{\Theta}(n)$ per message bit
- Key size: $\tilde{\Theta}(n^2)$
More on Regev’s encryption

- This scheme is homomorphic for addition: add ciphertexts
- And also for multiplication: tensor ciphertexts
  ⇒ Can be turned into FHE [Br12]

- Enc and KeyGen may be swapped: dual-Regev [GePeVa08]
  ⇒ This allows ID-based encryption, and more

May be turned into a practical scheme [Pe14]

- Use Ring-LWE rather than LWE: more efficient
- Ciphertext expansion can be lowered to essentially 1
- IND-CCA security can be achieved efficiently in the ROM
More on Regev’s encryption

- This scheme is homomorphic for addition: add ciphertexts
- And also for multiplication: tensor ciphertexts
  \[ \Rightarrow \text{Can be turned into FHE} \quad [Br12] \]
- Enc and KeyGen may be swapped: dual-Regev [GePeVa08]
  \[ \Rightarrow \text{This allows ID-based encryption, and more} \]
- May be turned into a practical scheme \quad [Pe14]
  - Use Ring-LWE rather than LWE: more efficient
  - Ciphertext expansion can be lowered to essentially 1
  - IND-CCA security can be achieved efficiently in the ROM
More on Regev’s encryption

- This scheme is homomorphic for addition: add ciphertexts
- And also for multiplication: tensor ciphertexts
  ⇒ Can be turned into FHE \([Br12]\)

- Enc and KeyGen may be swapped: dual-Regev \([GePeVa08]\)
  ⇒ This allows ID-based encryption, and more

May be turned into a practical scheme \([Pe14]\)

- Use Ring-LWE rather than LWE: more efficient
- Ciphertext expansion can be lowered to essentially 1
- IND-CCA security can be achieved efficiently in the ROM
Road-map

- Definition of the LWE problem
- Regev’s encryption scheme
- **Lattice problems**
- Hardness of LWE
- Equivalent problems
Euclidean lattices

Lattice $L = \sum_{i=1}^{n} \mathbb{Z}b_i \subset \mathbb{R}^n$, for some linearly indep. $b_i$’s.

Minimum $\lambda(L) = \min (\|b\| : b \in L \setminus 0)$.

$\text{SVP}_\gamma$: Given as input a basis of $L$, find $b \in L$ s.t. $0 < \|b\| \leq \gamma \cdot \lambda(L)$.

$\text{BDD}_\gamma$: Given as input a basis of $L$, and a vector $t$ s.t. $\text{dist}(t, L) < \frac{1}{2\gamma} \cdot \lambda(L)$, find $b \in L$ minimizing $\|b - t\|$. 
Euclidean lattices

Lattice \( L = \sum_{i=1}^{n} \mathbb{Z}b_i \subset \mathbb{R}^n \), for some linearly indep. \( b_i \)'s.

Minimum \( \lambda(L) = \min (\|b\| : b \in L \setminus \{0\}) \).

**SVP\( \gamma \):** Given as input a basis of \( L \), find \( b \in L \) s.t. \( 0 < \|b\| \leq \gamma \cdot \lambda(L) \).

**BDD\( \gamma \):** Given as input a basis of \( L \), and a vector \( t \) s.t. \( \text{dist}(t, L) < \frac{1}{2\gamma} \cdot \lambda(L) \), find \( b \in L \) minimizing \( \|b - t\| \).
Euclidean lattices

Lattice $L = \sum_{i=1}^{n} \mathbb{Z}b_i \subset \mathbb{R}^n$, for some linearly indep. $b_i$’s.

Minimum $\lambda(L) = \min (\|b\| : b \in L \setminus \{0\})$.

$\text{SVP}_\gamma$: Given as input a basis of $L$, find $b \in L$ s.t. $0 < \|b\| \leq \gamma \cdot \lambda(L)$.

$\text{BDD}_\gamma$: Given as input a basis of $L$, and a vector $t$ s.t. $\text{dist}(t, L) < \frac{1}{2\gamma} \cdot \lambda(L)$, find $b \in L$ minimizing $\|b - t\|$.
Best known (classical/quantum) algorithms

**SVP** $\gamma$: Given $L$, find $b \in L$ s.t. $0 < \|b\| \leq \gamma \cdot \lambda(L)$.

**BDD** $\gamma$: Given $L$ and $t \in \mathbb{R}^n$ s.t. $\text{dist}(t, L) < \frac{1}{2\gamma} \cdot \lambda(L)$, find $b \in L$ minimizing $\|b - t\|$.

For small $\gamma$: [AgDaReSD15]
- Time $2^{n/2}$.
- In practice: up to $n \approx 120$ (with other algorithms).

For $\gamma = n^{\Omega(1)}$: [ScEu91, HaPuSt11]
- Time $\left(\frac{n}{\log \gamma}\right)^{O\left(\frac{n}{\log \gamma}\right)}$.
- In practice, we can reach $\gamma \approx 1.01^n$ [ChNg11].

https://github.com/dstehle/fplll
Best known (classical/quantum) algorithms

\textbf{SVP}_\gamma: \text{ Given } L, \text{ find } b \in L \text{ s.t. } 0 < \|b\| \leq \gamma \cdot \lambda(L).

\textbf{BDD}_\gamma: \text{ Given } L \text{ and } t \in \mathbb{R}^n \text{ s.t. } \text{dist}(t, L) < \frac{1}{2\gamma} \cdot \lambda(L),
\text{ find } b \in L \text{ minimizing } \|b - t\|.

For small \( \gamma \): \cite{AgDaReSD15}

- Time \( 2^{n/2} \).
- In practice: up to \( n \approx 120 \) (with other algorithms).

For \( \gamma = n^{\Omega(1)} \): \cite{ScEu91, HaPuSt11}

- Time \( \left( \frac{n}{\log \gamma} \right)^{O\left( \frac{n}{\log \gamma} \right)} \).
- In practice, we can reach \( \gamma \approx 1.01^n \) \cite{ChNg11}.

https://github.com/dstehle/fplll
Best known (classical/quantum) algorithms

**SVP**$\gamma$: Given $L$, find $b \in L$ s.t. $0 < \|b\| \leq \gamma \cdot \lambda(L)$.

**BDD**$\gamma$: Given $L$ and $t \in \mathbb{R}^n$ s.t. $\text{dist}(t, L) < \frac{1}{2\gamma} \cdot \lambda(L)$, find $b \in L$ minimizing $\|b - t\|$.

For small $\gamma$:  
- Time $2^{n/2}$.
- In practice: up to $n \approx 120$ (with other algorithms).

For $\gamma = n^{\Omega(1)}$:  
- BKZ [ScEu91,HaPuSt11]
  - Time $\left(\frac{n}{\log \gamma}\right)^O\left(\frac{n}{\log \gamma}\right)$.
  - In practice, we can reach $\gamma \approx 1.01^n$ [ChNg11].

https://github.com/dstehle/fplll
GapSVP$_\gamma$

Given a basis of a lattice $L$ and $d > 0$, assess whether

$$\lambda(L) \leq d \quad \text{or} \quad \lambda(L) > \gamma \cdot d.$$ 

- **NP-hard** when $\gamma \leq O(1)$ (random. red.) [Aj98,HaRe07]
- **In \ NP \cap \text{coNP}** when $\gamma \geq \sqrt{n}$ [GoGo98,AhRe04]
- **In \ P** when $\gamma \geq \exp \left( n \cdot \frac{\log \log n}{\log n} \right)$ (BKZ)
Hardness of SVP

GapSVP\(_\gamma\)

Given a basis of a lattice \(L\) and \(d > 0\), assess whether

\[
\lambda(L) \leq d \quad \text{or} \quad \lambda(L) > \gamma \cdot d.
\]

- **NP-hard** when \(\gamma \leq O(1)\) (random. red.) \([Aj98,HaRe07]\)
- In **NP\cap coNP** when \(\gamma \geq \sqrt{n}\) \([GoGo98,AhRe04]\)
- In **P** when \(\gamma \geq \exp\left(n \cdot \frac{\log \log n}{\log n}\right)\) \((BKZ)\)
Road-map

- Definition of the LWE problem
- Regev’s encryption scheme
- Lattice problems
- **Hardness of LWE**
- Equivalent problems

Each LWE sample gives $\approx \log_2 \frac{1}{\alpha}$ bits of data on secret $s$.

With a few samples, $s$ is uniquely specified. How to find it?
Exhaustive search

Assume we are given $A$ and $b = As + e$, for some $e$ whose entries are $\approx \alpha q$.
We want to find $s$.

1st variant:
- Try all the possible $s \in \mathbb{Z}_q^n$.
- Test if $b - A \cdot s$ is small.

$\Rightarrow$ Cost $\approx q^n$.

2nd variant:
- Try all the possible $n$ first error terms.
- Recover the corresponding $s$, by linear algebra.
- Test if $b - A \cdot s$ is small.

$\Rightarrow$ Cost $\approx (\alpha q \sqrt{\log n})^n$. 
Assume we are given $A$ and $b = As + e$, for some $e$ whose entries are $\approx \alpha q$. We want to find $s$.

1st variant:
- Try all the possible $s \in \mathbb{Z}_{q}^{n}$.
- Test if $b - A \cdot s$ is small.

$\Rightarrow$ Cost $\approx q^{n}$.

2nd variant:
- Try all the possible $n$ first error terms.
- Recover the corresponding $s$, by linear algebra.
- Test if $b - A \cdot s$ is small.

$\Rightarrow$ Cost $\approx (\alpha q \sqrt{\log n})^{n}$.
Solving LWE with BKZ (1/2)

Assume we are given $A$ and $b = As + e$, for some $e$ whose entries are $\approx \alpha q$. We want to find $s$.

Let $L_A = \{x \in \mathbb{Z}^m : \exists s \in \mathbb{Z}^n, x = As \lfloor q \rfloor\} = AZ_q^n + q\mathbb{Z}^m$.

- $L_A$ is a lattice of dimension $m$.
- Whp, its minimum satisfies $\lambda(L) \approx \sqrt{m} \cdot q^{1-\frac{n}{m}}$.
- We have $\text{dist}(b, L) = \|e\| \approx \sqrt{m} \alpha q$.

**LWE reduces to BDD**

This is a BDD instance in dim $m$ with $\gamma \approx q^{-\frac{n}{m}}/\alpha$. 
Solving LWE with BKZ (1/2)

Assume we are given $A$ and $b = As + e$, for some $e$ whose entries are $\approx \alpha q$. We want to find $s$.

Let $L_A = \{ x \in \mathbb{Z}^m : \exists s \in \mathbb{Z}^n, x = As \ [q] \} = A\mathbb{Z}_q^n + q\mathbb{Z}^m$.

- $L_A$ is a lattice of dimension $m$.
- Whp, its minimum satisfies $\lambda(L) \approx \sqrt{m} \cdot q^{1 - \frac{n}{m}}$.
- We have $\text{dist}(b, L) = \|e\| \approx \sqrt{m} \alpha q$.

LWE reduces to BDD

This is a BDD instance in dim $m$ with $\gamma \approx q^{-\frac{n}{m}} / \alpha$. 
Solving LWE with BKZ (1/2)

Assume we are given $A$ and $b = As + e$, for some $e$ whose entries are $\approx \alpha q$. We want to find $s$.

Let $L_A = \{x \in \mathbb{Z}^m : \exists s \in \mathbb{Z}^n, x = As \pmod{q}\} = A\mathbb{Z}_q^n + q\mathbb{Z}^m$.

- $L_A$ is a lattice of dimension $m$.
- Whp, its minimum satisfies $\lambda(L) \approx \sqrt{m} \cdot q^{1-n/m}$.
- We have $\text{dist}(b, L) = \|e\| \approx \sqrt{m\alpha q}$.

LWE reduces to BDD

This is a BDD instance in dim $m$ with $\gamma \approx q^{-n/m}/\alpha$. 
Assume we are given $A$ and $b = As + e$, for some $e$ whose entries are $\approx \alpha q$. We want to find $s$.

Let $L_A = \{ x \in \mathbb{Z}^m : \exists s \in \mathbb{Z}^n, x = As \ [q] \} = A\mathbb{Z}_q^n + q\mathbb{Z}^m$.

- $L_A$ is a lattice of dimension $m$.
- Whp, its minimum satisfies $\lambda(L) \approx \sqrt{m} \cdot q^{1-\frac{n}{m}}$.
- We have $\text{dist}(b, L) = \|e\| \approx \sqrt{m\alpha q}$.

**LWE reduces to BDD**

This is a BDD instance in dim $m$ with $\gamma \approx q^{-\frac{n}{m}}/\alpha$. 
Solving LWE with BKZ (2/2)

**LWE reduces to BDD**

This is a BDD instance in dim $m$ with $\gamma \approx q^{-\frac{n}{m}}/\alpha$.

Cost of BKZ: $(\frac{m}{\log \gamma})^{O(\frac{m}{\log \gamma})}$, with $
\frac{\log \gamma}{m} = \frac{1}{m} \log \frac{1}{\alpha} - \frac{n \log q}{m^2}$.

Cost is minimized for $m \approx \frac{2n \log q}{\log \frac{1}{\alpha}}$.

**Cost of BKZ to solve LWE**

Time: $(\frac{n \log q}{\log^2 \alpha})^{O(\frac{n \log q}{\log^2 \alpha})}$. 
Solving LWE with BKZ (2/2)

LWE reduces to BDD

This is a BDD instance in dim $m$ with $\gamma \approx q^{-n/m}/\alpha$.

Cost of BKZ: $(\frac{m}{\log \gamma})^{O\left(\frac{m}{\log \gamma}\right)}$, with $\frac{\log \gamma}{m} = \frac{1}{m} \log \frac{1}{\alpha} - \frac{n \log q}{m^2}$.

Cost is minimized for $m \approx \frac{2n \log q}{\log \frac{1}{\alpha}}$.

Cost of BKZ to solve LWE

Time: $\left(\frac{n \log q}{\log^2 \alpha}\right)^{O\left(\frac{n \log q}{\log^2 \alpha}\right)}$. 
Assume that $\alpha q \geq 2\sqrt{n}$.

[Re05]

If $q$ is prime and $\leq n^{O(1)}$, then there exists a quantum polynomial-time reduction from $\text{SVP}_\gamma$ in $\text{dim } n$ to $\text{LWE}_{n,q,\alpha}$, with $\gamma \approx n/\alpha$.

[BrLaPeReSt13]

If $q \leq n^{O(1)}$, then there exists a classical polynomial-time reduction from $\text{GapSVP}_\gamma$ in $\text{dim } \sqrt{n}$ to $\text{LWE}_{n,q,\alpha}$, with $\gamma \approx n/\alpha$.

- The two results are incomparable.
- Best achievable $\gamma$ here: $n$.
- In the case of Regev’s encryption, we get $\gamma \approx n^{3/2}$.
- One can use $\text{BDD}_\gamma$ instead (with a different $\gamma$).
Assume that $\alpha q \geq 2\sqrt{n}$.

[Re05]
If $q$ is prime and $\leq n^O(1)$, then there exists a quantum polynomial-time reduction from $\text{SVP}_\gamma$ in $\text{dim } n$ to $\text{LWE}_{n,q,\alpha}$, with $\gamma \approx n/\alpha$.

[BrLaPeReSt13]
If $q$ is $\leq n^O(1)$, then there exists a classical polynomial-time reduction from $\text{GapSVP}_\gamma$ in $\text{dim } \sqrt{n}$ to $\text{LWE}_{n,q,\alpha}$, with $\gamma \approx n/\alpha$.

- The two results are incomparable.
- Best achievable $\gamma$ here: $n$.
- In the case of Regev's encryption, we get $\gamma \approx n^{3/2}$.
- One can use $\text{BDD}_\gamma$ instead (with a different $\gamma$).
Assume that $\alpha q \geq 2\sqrt{n}$.

**[Re05]**

If $q$ is prime and $\leq n^{O(1)}$, then there exists a *quantum* polynomial-time reduction from $\text{SVP}_\gamma$ in $\text{dim } n$ to $\text{LWE}_{n,q,\alpha}$, with $\gamma \approx \frac{n}{\alpha}$.

**[BrLaPeReSt13]**

If $q$ is $\leq n^{O(1)}$, then there exists a *classical* polynomial-time reduction from $\text{GapSVP}_\gamma$ in $\text{dim } \sqrt{n}$ to $\text{LWE}_{n,q,\alpha}$, with $\gamma \approx \frac{n}{\alpha}$.

- The two results are incomparable.
- Best achievable $\gamma$ here: $n$.
- In the case of Regev’s encryption, we get $\gamma \approx n^{3/2}$.
- One can use $\text{BDD}_\gamma$ instead (with a different $\gamma$).
Road-map

- Definition of the LWE problem
- Regev’s encryption scheme
- Lattice problems
- Hardness of LWE
- Equivalent problems
LWE variants

Numerous variants have been showed to be at least as hard as LWE, up to polynomial factors in the noise rate $\alpha$:

(Polynomial in $n$, $\log q$ and possibly in the number of samples $m$.)

- When $s$ is distributed from the error distribution.
- When $s$ is binary with sufficient entropy.
- When $e$ is uniform in a hypercube.
- When $e$ corresponds to a deterministic rounding of $As$.
- When $A$ is binary (modulo $q$).
- When some extra information on $e$ is provided.
- When the first component of $e$ is zero.
**LWE in dimension 1**

1-dimensional LWE \[\text{[BoVe96]}\]

With non-negl. prob. over \(s \leftarrow U(\mathbb{Z}_q)\): distinguish between

\[(a, a \cdot s + e) \text{ and } (a, b) \quad (\text{over } \mathbb{Z}_q^2),\]

where \(a, b \leftarrow U(\mathbb{Z}_q), e \leftarrow D_{\mathbb{Z},\alpha q}\).

Hardness of 1-dim LWE \[\text{[BrLaPeReSt13]}\]

For any \(n, q, n', q'\) with \(n \log q \leq n' \log q'\):

there exists a polynomial-time reduction from LWE\(_{n,q,\alpha}\) to LWE\(_{n',q',\alpha'}\) for some \(\alpha' \leq \alpha \cdot (n \log q)^{O(1)}\).

\[\Rightarrow \text{LWE}_{1,q^n} \text{ is no easier than LWE}_{n,q}.\]
Approximate gcd

**AGCD\(_{D,N,\alpha}\) [HG01]**

With non-negl. prob. over \(p \leftarrow D\), distinguish between

\[u \quad \text{and} \quad q \cdot p + r \quad \text{(over } \mathbb{Z}),\]

where \(u \leftarrow U([0, N))\), \(q \leftarrow U([0, \frac{N}{p}))\), \(r \leftarrow \lfloor D_\alpha p \rfloor\).

**Hardness of AD (Informal) [ChSt15]**

AGCD\(_{D,N,\alpha}\) is computationally equivalent to LWE\(_{n,q,\alpha}\), for some \(D\) of mean \(\approx q^n\) and some \(N \approx q^{2n}\).
Conclusion

LWE:
- LWE is hard for almost all instances.
- It seems exponentially hard to solve, even quantumly.
- It is a rich/expressive problem, convenient for cryptographic design.

Lattices:
- LWE hardness comes from lattice problems.
- We can design lattice-based cryptosystems without knowing lattices!
Exciting topics I did not mention

- The Small Integer Solution problem (SIS) ⇒ Digital signatures.
- Ideal lattices, Ring-LWE, Ring-SIS, NTRU ⇒ Using polynomial rings (a.k.a. structured matrices) to get more efficient constructions.
- Implementation of lattice-based primitives.

These will be addressed in Léo’s talk (Friday morning), my second talk (Friday afternoon) and Tim’s talk (Friday afternoon).
Open problems: foundations

If $q$ is prime and $\leq n^{O(1)}$, then there exists a **quantum** polynomial-time reduction from $\text{SVP}_\gamma$ in $\text{dim } n$ to $\text{LWE}_{n,q,\alpha}$, with $\gamma \approx n/\alpha$.

If $q$ is $\leq n^{O(1)}$, then there exists a **classical** polynomial-time reduction from $\text{GapSVP}_\gamma$ in $\text{dim } \sqrt{n}$ to $\text{LWE}_{n,q,\alpha}$, with $\gamma \approx n/\alpha$.

- Does there exist a classical reduction from $n$-dimensional $\text{SVP}_\gamma/\text{BDD}_\gamma$ to $\text{LWE}_{n,q,\alpha}$?
- Does there exist a quantum algorithm for $\text{LWE}_{n,q,\alpha}$ that runs in time $2^{\sqrt{n}}$ (when $q \leq n^{O(1)}$)?
- Is LWE easy for some $\alpha = 1 / n^{O(1)}$?
- Can we reduce factoring/DL to LWE?
Open problems: foundations

If $q$ is prime and $\leq n^{O(1)}$, then there exists a **quantum** polynomial-time reduction from $\text{SVP}_\gamma$ in $\text{dim } n$ to $\text{LWE}_{n,q,\alpha}$, with $\gamma \approx n/\alpha$.

If $q$ is $\leq n^{O(1)}$, then there exists a **classical** polynomial-time reduction from $\text{GapSVP}_\gamma$ in $\text{dim } \sqrt{n}$ to $\text{LWE}_{n,q,\alpha}$, with $\gamma \approx n/\alpha$.

- Does there exist a classical reduction from $n$-dimensional $\text{SVP}_\gamma/\text{BDD}_\gamma$ to $\text{LWE}_{n,q,\alpha}$?
- Does there exist a quantum algorithm for $\text{LWE}_{n,q,\alpha}$ that runs in time $2^{\sqrt{n}}$ (when $q \leq n^{O(1)}$)?
- Is $\text{LWE}$ easy for some $\alpha = 1/n^{O(1)}$?
- Can we reduce factoring/DL to $\text{LWE}$?
Open problems: foundations

If $q$ is prime and $\leq n^{O(1)}$, then there exists a **quantum** polynomial-time reduction from $\text{SVP}_\gamma$ in $\text{dim } n$ to $\text{LWE}_{n,q,\alpha}$, with $\gamma \approx n/\alpha$.

If $q$ is $\leq n^{O(1)}$, then there exists a **classical** polynomial-time reduction from $\text{GapSVP}_\gamma$ in $\text{dim } \sqrt{n}$ to $\text{LWE}_{n,q,\alpha}$, with $\gamma \approx n/\alpha$.

- Does there exist a classical reduction from $n$-dimensional $\text{SVP}_\gamma/\text{BDD}_\gamma$ to $\text{LWE}_{n,q,\alpha}$?
- Does there exist a quantum algorithm for $\text{LWE}_{n,q,\alpha}$ that runs in time $2^{\sqrt{n}}$ (when $q \leq n^{O(1)}$)?
- Is LWE easy for some $\alpha = 1/n^{O(1)}$?
- Can we reduce factoring/DL to LWE?
Open problems: foundations

If $q$ is prime and $\leq n^{O(1)}$, then there exists a **quantum** polynomial-time reduction from $\text{SVP}_\gamma$ in $\text{dim } n$ to $\text{LWE}_{n,q,\alpha}$, with $\gamma \approx n/\alpha$.

If $q$ is $\leq n^{O(1)}$, then there exists a **classical** polynomial-time reduction from $\text{GapSVP}_\gamma$ in $\text{dim } \sqrt{n}$ to $\text{LWE}_{n,q,\alpha}$, with $\gamma \approx n/\alpha$.

- Does there exist a classical reduction from $n$-dimensional $\text{SVP}_\gamma/\text{BDD}_\gamma$ to $\text{LWE}_{n,q,\alpha}$?
- Does there exist a quantum algorithm for $\text{LWE}_{n,q,\alpha}$ that runs in time $2^{\sqrt{n}}$ (when $q \leq n^{O(1)}$)?
- Is LWE easy for some $\alpha = 1 / n^{O(1)}$?
- Can we reduce factoring/DL to LWE?
Open problems: cryptanalysis

LWE-based cryptography is based on $\text{GapSVP}_\gamma$ for $\gamma \geq n$. No NP-hardness here...

- Can we solve $\text{SVP}_\gamma$ in $\text{poly}(n)$-time for some $\gamma = n^{O(1)}$?
- And with a quantum computer?
- Can we do better than BKZ’s $(\frac{n}{\log \gamma})^{O(\frac{n}{\log \gamma})}$ run-time, for some $\gamma$?
- What are the practical limits?

http://www.latticechallenge.org
Open problems: cryptanalysis

LWE-based cryptography is based on $\text{GapSVP}_\gamma$ for $\gamma \geq n$.
No NP-hardness here...

- Can we solve $\text{SVP}_\gamma$ in $\text{poly}(n)$-time for some $\gamma = n^{O(1)}$?
- And with a quantum computer?
- Can we do better than BKZ’s $(\frac{n}{\log \gamma})^{O\left(\frac{n}{\log \gamma}\right)}$ run-time, for some $\gamma$?
- What are the practical limits?

http://www.latticechallenge.org
Open problems: cryptanalysis

LWE-based cryptography is based on $\text{GapSVP}_\gamma$ for $\gamma \geq n$. No NP-hardness here...

- Can we solve $\text{SVP}_\gamma$ in $\text{poly}(n)$-time for some $\gamma = n^{O(1)}$?
- And with a quantum computer?
- Can we do better than BKZ’s $\left(\frac{n}{\log \gamma}\right)^{O\left(\frac{n}{\log \gamma}\right)}$ run-time, for some $\gamma$?
- What are the practical limits?

http://www.latticechallenge.org
Open problems: cryptanalysis

LWE-based cryptography is based on \( \text{GapSVP}_\gamma \) for \( \gamma \geq n \).
No NP-hardness here...

- Can we solve \( \text{SVP}_\gamma \) in \( \text{poly}(n) \)-time for some \( \gamma = n^{O(1)} \)?
- And with a quantum computer?
- Can we do better than BKZ’s \( \left( \frac{n}{\log \gamma} \right)^{O\left(\frac{n}{\log \gamma}\right)} \) run-time, for some \( \gamma \)?
- What are the practical limits?

http://www.latticechallenge.org
Open problems: practice

There exist practical lattice-based signature and encryption schemes.

- Can lattice-based primitives outperform other approaches in some contexts?
- What about side-channel cryptanalysis?
- Can advanced lattice-based primitives be made practical? Attribute-based encryption? Homomorphic encryption?
Open problems: practice

There exist practical lattice-based signature and encryption schemes.

- Can lattice-based primitives outperform other approaches in some contexts?
- What about side-channel cryptanalysis?
- Can advanced lattice-based primitives be made practical? Attribute-based encryption? Homomorphic encryption?
Open problems: practice

There exist practical lattice-based signature and encryption schemes.

- Can lattice-based primitives outperform other approaches in some contexts?
- What about side-channel cryptanalysis?
- Can advanced lattice-based primitives be made practical?

Attribute-based encryption? Homomorphic encryption?
Open problems: practice

There exist practical lattice-based signature and encryption schemes.

- Can lattice-based primitives outperform other approaches in some contexts?
- What about side-channel cryptanalysis?
- Can advanced lattice-based primitives be made practical?
  Attribute-based encryption? Homomorphic encryption?


AgDaReSD15  D. Aggarwal, D. Dadush, O. Regev, N. Stephens-Davidowitz: Solving the Shortest Vector Problem in $2^n$ Time via Discrete Gaussian Sampling. Available on ARXIV.


## Bibliography

<table>
<thead>
<tr>
<th>Reference</th>
<th>Authors</th>
<th>Title</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>GoVaWe13</td>
<td>S. Gorbunov, V. Vaikuntanathan, H. Wee</td>
<td>Attribute-based encryption for circuits. STOC 2013: 545-554.</td>
<td></td>
</tr>
</tbody>
</table>