An Introduction to the Learning with Errors Problem in 3 Hours

賴奕甫

Peter W. Shor

Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*

Peter W. Shor†

Abstract

A digital computer is generally believed to be an efficient universal computing device; that is, it is believed able to simulate any physical computing device with an increase in computation time by at most a polynomial factor. This may not be true when quantum mechanics is taken into consideration. This paper considers factoring integers and finding discrete logarithms, two problems which are generally thought to be hard on a classical computer and which have been used as the basis of several proposed cryptosystems. Efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms take a number of steps polynomial in the input size, e.g., the number of digits of the integer to be factored.

Keywords: algorithmic number theory, prime factorization, discrete logarithms, Church's thesis, quantum computers, foundations of quantum mechanics, spin systems, Fourier transforms

AMS subject classifications: 81P10, 11Y05, 68Q10, 03D10

There are polynomial-time quantum algorithms that can solve

- Factorization Problem
- Discrete Logarithm Problem (over $\mathbb{Z}_p$)

About ECC

Shor's discrete logarithm quantum algorithm for elliptic curves

John Proos and Christof Zalka

Department of Combinatorics and Optimization
University of Waterloo, Waterloo, Ontario
Canada N2L 3G1
e-mail: japroos@math.uwaterloo.ca zalka@iqc.ca

February 1, 2008

Abstract

We show in some detail how to implement Shor's efficient quantum algorithm for discrete logarithms for the particular case of elliptic curve groups. It turns out that for this problem a smaller quantum computer can solve problems further beyond current computing than for integer factorisation. A 160 bit elliptic curve cryptographic key could be broken on a quantum computer using around 1000 qubits while factoring the security-wise equivalent 1024 bit RSA modulus would require about 2000 qubits. In this paper we only consider elliptic curves over GF(p) and not yet the equally important ones over GF(2^n) or other finite fields. The main technical difficulty is to implement Euclid's gcd algorithm to compute multiplicative inverses modulo p. As the runtime of Euclid's algorithm depends on the input, one difficulty encountered is the "quantum halting problem".

There is also a quantum algorithm that can solve

- Discrete Logarithm Problem over Elliptic Curves

; (

If there was a practical quantum computer, then it was able to break

- RSA encryption, signature scheme
- Diffie-Hellman Key Exchange, Elgamal, DSA
- Elliptic Curve Diffie-Hellman (ECDH), Elliptic Curve DSA (ECDSA)
- …etc

Transport Layer Security

Key exchange or key agreement [edit]

Before a client and server can begin to exchange information protected by TLS, they must securely exchange or agree upon an encryption key and a cipher to use when encrypting data (see Cipher). Among the methods used for key exchange/agreement are: public and private keys generated with RSA (denoted TLS_RSA in the TLS handshake protocol), Diffie-Hellman (TLS_DH), ephemeral Diffie-Hellman (TLS_DHE), Elliptic Curve Diffie-Hellman (TLS_ECDH), ephemeral Elliptic Curve Diffie-Hellman (TLS_ECDHE), anonymous Diffie-Hellman (TLS_DH_anon),[1] pre-shared key (TLS_PSK)[29] and Secure Remote Password (TLS_SRP).[30]

No more forward secrecy.

Post-Quantum Cryptography

- Post-quantum cryptography is a branch of cryptography that considers cryptographic algorithms which is still secure against quantum attack.

- Lattice-based cryptography
- Multivariate cryptography
- Hash-based cryptography
- Code-based cryptography
- Supersingular elliptic curve isogeny cryptography

Workshop on Cybersecurity in a Post-Quantum World

The advent of practical quantum computing will break all commonly u
algorithms. In response, NIST is researching cryptographic algorithms f
agreement and digital signatures that are not susceptible to cryptanaly
NIST is holding this workshop to engage academic, industry, and gover
Quantum Workshop will be held on April 2-3, 2015, immediately follow
Conference on Practice and Theory of Public-Key Cryptography. NIST se
post-quantum cryptography and its potential future standardization.

PQCRYPTO
ICT-645622

Horizon 2020

PQCRYPTO

Post-Quantum Cryptography for Long-Term Security

Project number: Horizon 2020 ICT-645622

Initial recommendations of long-term secure post-quantum systems

Due date of deliverable: none
Actual submission date: 7. September 2015

LWE Problem

- The *learning with errors problem* (LWE) is included in the lattice-based cryptography.
- The LWE problem is versatile (it can be used to construct a variety of cryptographic algorithms). For example,

  PKE CPA-secure [Reg05, PVW08, LP11]

  PKE CCA-secure[PW08, Pei09, MP12]

  Oblivious transfer[PVW08]

  Identity-based encryption[GPV08, CHKP10, ABB10a, ABB10b]

  Fully Homomorphic Encryption[BV11, BGV12]

A Simple Introduction to the Lattice

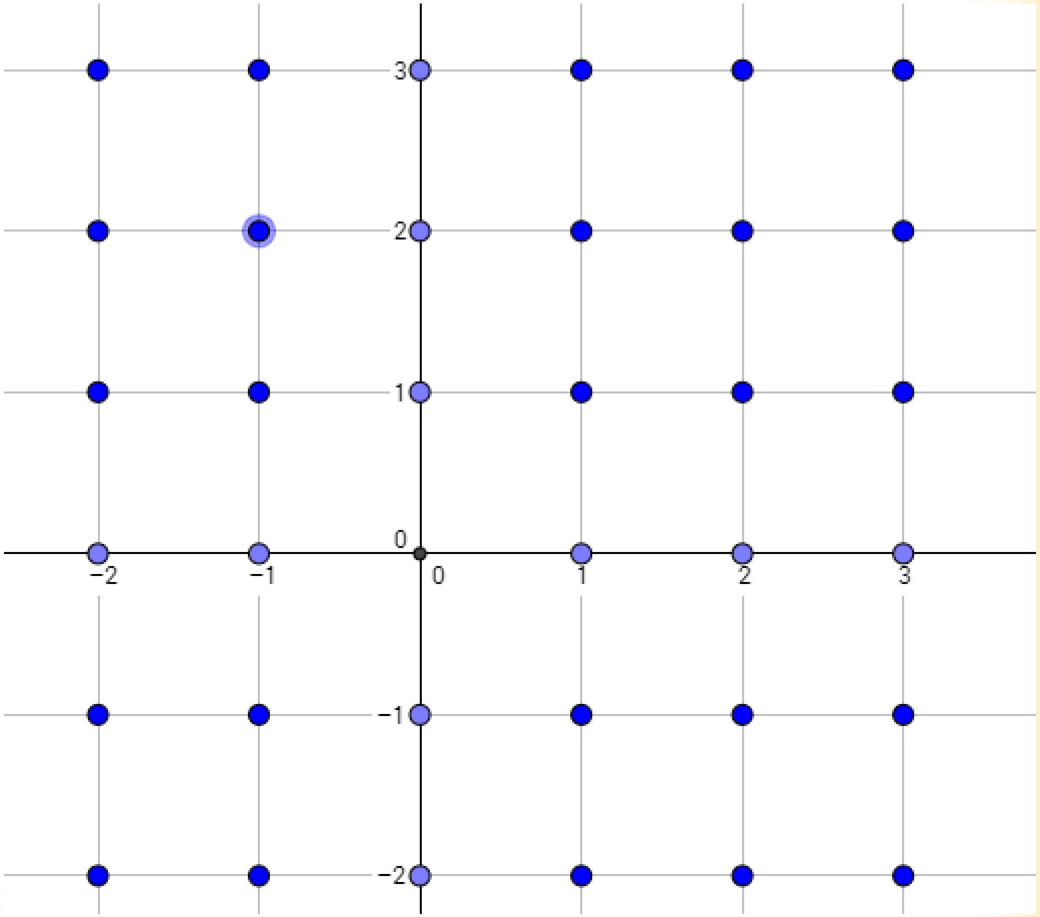
- This is a lattice in $\mathbb{R}^2$.

It can be written as…

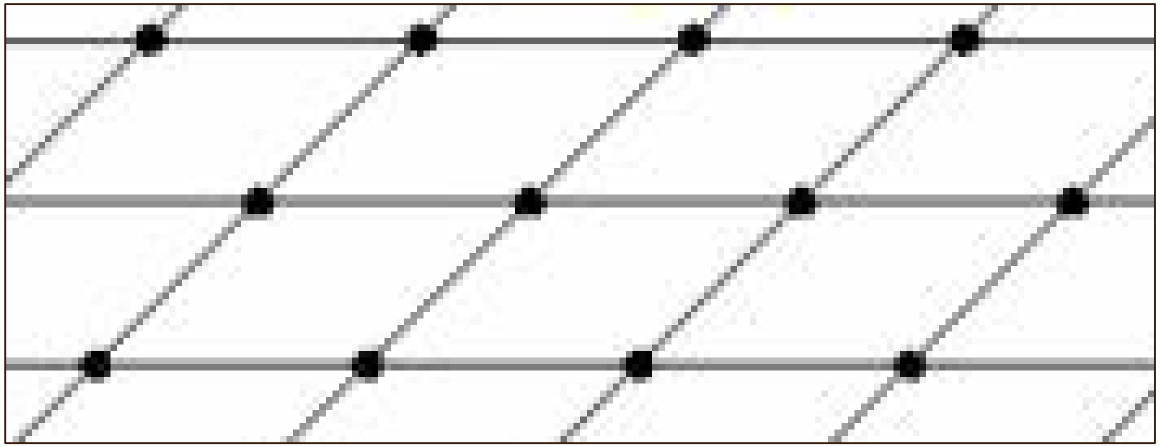
- $\{(x, y) | x, y \in \mathbb{Z}\}$

Also written as…

- $\begin{pmatrix}1\\0\end{pmatrix}\mathbb{Z} + \begin{pmatrix}0\\1\end{pmatrix}\mathbb{Z}$
- $\begin{pmatrix}1\\1\end{pmatrix}\mathbb{Z} + \begin{pmatrix}0\\1\end{pmatrix}\mathbb{Z}$
- $\begin{pmatrix}1\\0\end{pmatrix}\mathbb{Z} + \begin{pmatrix}1\\1\end{pmatrix}\mathbb{Z}$
- …



A Lattice



A Simple Introduction to the Lattice

- **Definition 1.** For a linear independent set $B = \{u_1, \dots, u_k\} \subset \mathbb{R}^n$, a lattice L generated by B in $\mathbb{R}^n$ is defined to be
$$L = \sum \mathbb{Z}u_i \,.$$

(If $k = n$, then the lattice is said to be a full-rank lattice.)

There is an equivalent definition for the lattice in $\mathbb{R}^n$--

A Simple Introduction to the Lattice

- **Definition 2.** A set $L \subset \mathbb{R}^n$ is said to be a *discrete additive subgroup* if it satisfies the following two conditions:
  1. It is closed under addition and substraction. (additive subgroup)
  2. There is a constant $\epsilon > 0$ such that for any $v \in L$

$$L \cap \{w \in \mathbb{R}^n : ||v - w||\} = \{v\}.$$ (discrete)

- **Theorem.** In $\mathbb{R}^n$, a subset of $\mathbb{R}^n$ is a lattice if and only if it is a discrete additive subgroup.

(see p.25 in *Algebraic Number Theory* by Neukirch)

Mythology in Lattice-Based Cryptography

- Given a linear independent generating set of a lattice L in $\mathbb{R}^n$.
- *Closest vector problem* (CVP):

  Given: a vector w in $\mathbb{R}^n$.

  Request: Find $v \in L$ such that

$$\|w - v\| = \min_{u \in L} \|w - u\|$$

- *Shortest vector problem* (SVP):

  Request: Find a nonzero vector $v \in L$ such that

$$\|v\| = \min_{u \in L - \{0\}} \|u\|$$

Hardness:

[Ajt98] M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract). In *STOC*, pages 10–19. 1998.

CVP≥SVP (sketch)

- Given a lattice L generated by a independent set $\{b_1, \dots, b_n\}$.
- Write the shortest nonzero lattice point $v = \sum a_i b_i$.

(Note that $a_i \in \mathbb{Z}$ can not be all even.)

- For each i, feed the CVP oracle with (L_i', b_i) where lattice L_i' is generated by $\{b_1, \dots, b_{i-1}, 2b_i, b_{i+1}, b_n\}$
- And the output is v_i

- Then $v_j - b_j$ is the shortest nonzero vector for some j.

(Why?)

Linear Equations

- There is a secret vector $\boldsymbol{s} = (s_1, s_2, s_3, s_4)^T \in \mathbb{Z}_{13}^4$

- Given

$$1s_1 + 2s_2 + 5s_3 + 2s_4 = 9 \pmod{13}$$
$$12s_1 + 1s_2 + 1s_3 + 6s_4 = 7 \pmod{13}$$
$$6s_1 + 10s_2 + 3s_3 + 6s_4 = 1 \pmod{13}$$
$$10s_1 + 4s_2 + 12s_3 + 8s_4 = 0 \pmod{13}.$$

- Solve for $\boldsymbol{s}$

The Learning with Errors Problem (Search)

- There is a secret vector $\boldsymbol{s} = (s_1, s_2, s_3, s_4)^T \in \mathbb{Z}_{13}^4$
- Given

$$
\begin{gathered}
5s_1 + 5s_2 + (-3)s_3 + 7s_4 \approx 6 \quad (mod\ 13) \\
-1s_1 + 1s_2 + 2s_3 + (-5)s_4 \approx (-4) \quad (mod\ 13) \\
(-3)s_1 + 3s_2 + 7s_3 + 4s_4 \approx 2 \quad (mod\ 13) \\
5s_1 + 4s_2 + (-1)s_3 + 2s_4 \approx (-5)(mod\ 13) \\
(-4)s_1 + 6s_2 + 3s_3 + (-2)s_4 \approx 5 \quad (mod\ 13) \\
(-2)s_1 + 3s_2 + 1s_3 + 6s_4 \approx (-3)(mod\ 13).
\end{gathered}
$$

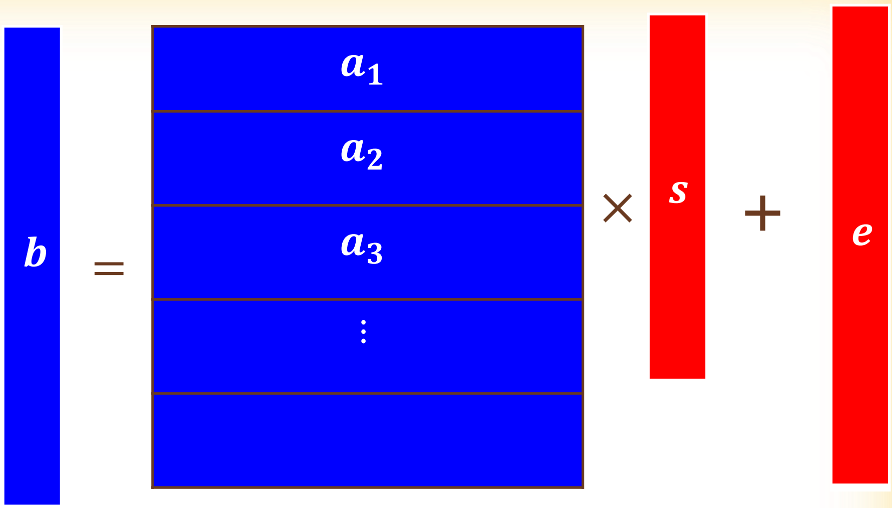
- There is an odds of ½ for each equation that is added by 1.
- Solve for $\boldsymbol{s}$

LWE distribution

- **(Definition)**

1. For a secret vector $\boldsymbol{s} \in \mathbb{Z}_q^n$ and distribution χ, an LWE distribution $\mathcal{A}_{\boldsymbol{s},n,q,\chi}$ generates a sample $(\boldsymbol{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ or $(A, \boldsymbol{b}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$ where $\boldsymbol{a}$ sampled uniformly from $\mathbb{Z}_q^n$ and $b = \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e$ where $e \leftarrow \chi$.

LWE Problem (Search)



- (Definition)

2. **LWE problem (Search):**
   - **Secret** $\boldsymbol{s} \in \mathbb{Z}_{\boldsymbol{q}}^{\boldsymbol{n}}$. Given $poly(n)$ LWE samples $(A, \boldsymbol{b})$ from $\mathcal{A}_{\boldsymbol{s},n,q,\chi}$.
   - Find $\boldsymbol{s}$.

$$
\begin{gathered}
5s_1 + 5s_2 + (-3)s_3 + 7s_4 \approx 6 \pmod{13} \\
-1s_1 + 1s_2 + 2s_3 + (-5)s_4 \approx (-4) \pmod{13} \\
(-3)s_1 + 3s_2 + 7s_3 + 4s_4 \approx 2 \pmod{13} \\
5s_1 + 4s_2 + (-1)s_3 + 2s_4 \approx (-5) (mod\ 13) \\
(-4)s_1 + 6s_2 + 3s_3 + (-2)s_4 \approx 5 \pmod{13} \\
(-2)s_1 + 3s_2 + 1s_3 + 6s_4 \approx (-3) (mod\ 13).
\end{gathered}
$$

LWE Problem (Search)

Dimension n
Modulus q
Error distribution χ

Adversary

Challenger

$(A, \boldsymbol{b})$

Output: $\boldsymbol{s} \in \mathbb{Z}_{\boldsymbol{q}}^{\boldsymbol{n}}$

Secret $\boldsymbol{s} \in \mathbb{Z}_{\boldsymbol{q}}^{\boldsymbol{n}}$

Generate :
$poly(n)$ LWE samples
$(A, \boldsymbol{b})$ from $\mathcal{A}_{\boldsymbol{s},n,q,\chi}$

LWE Problem (Decisional)

- (Definition)

3. **Decisional LWE problem:**

   - $\boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_{\boldsymbol{q}}^{\boldsymbol{n}}$. Given $poly(n)$ samples $(A, \boldsymbol{b})$ which are either from $\mathcal{A}_{\boldsymbol{s},n,q,\chi}$ or generated uniformly over $\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$ (with fair probabilities)
   - Determine which is the case in non-negligible advantage.

LWE Problem (Decisional)

Dimension n
Modulus q
Error distribution χ

Adversary

Challenger

$(A, \boldsymbol{b})$

Output: {“LWE”, ”uniform at random”}

$\boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_{\boldsymbol{q}}^{\boldsymbol{n}}$

Generate :
$poly(n)$ LWE samples $(A, \boldsymbol{b})$
either from $\mathcal{A}_{\boldsymbol{s},n,q,\chi}$ or
uniformly at random over
$\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$

For example

- Dimension $n = 4$
- Modulus $q = 13$
- Error distribution χ. ($\pm 1 \leftarrow \chi$ with prob ¼, $0 \leftarrow \chi$ with ½)

Given

$$\left(\begin{pmatrix} 0 & 6 & 0 & -2 & 5 \\ -6 & 6 & 0 & -2 & 4 \\ -5 & 6 & 6 & 3 & -3 \\ 4 & 3 & -2 & 1 & 1 \\ -5 & -3 & -5 & -4 & -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 7 \\ 4 \\ 9 \\ 0 \end{pmatrix} \right)$$

Q: LWE distribution or uniform distribution?

Remark.

- The distribution χ is called the "error distribution". χ is typically chosen to be a discrete Gaussian (normal) distribution with small standard deviation.
- The hardness varies with the S.D. of χ
- Oded Regev shows that $LWE \geq approx\text{-}S(I)VP$ & *$approx$–GapSVP with a quantum reduction.*
- Specifically, with $dimension\ n$, modulus $q = poly(n)$, and error distribution (discrete Gaussian distribution) χ of standard deviation αq,

$$LWE_{n,q,\chi} \geq (n/\alpha)\text{-}S(I)VP.$$

Little Knowledge:
How to check candidate $\boldsymbol{s}$?

- Given $poly(n)$ LWE samples $(A, \boldsymbol{b} = A\boldsymbol{s} + \boldsymbol{e}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ for some fixed secret $\boldsymbol{s} \in \mathbb{Z}_q^n$.

- Assume you find an algorithm that can generate a small set of candidate answers. How can you check which one may be the correct one?

A Little Question

For any $c \in \mathbb{Z}_q$. χ is some distribution over $\mathbb{Z}_q$.

$\Pr_{\substack{\mathrm{b} \xleftarrow{\$} \mathbb{Z}_q \\ a \leftarrow \chi}} [a + b = c] =$? ($a$ and b are generated independently)

If the modulus q is $poly(n)$-bounded, then DLWE=SLWE problem

"≤"

- Given $poly(n)$ samples either from a LWE distribution $\mathcal{A}_{\boldsymbol{s},n,q,\chi}$ for some unknown $\boldsymbol{s} \in \mathbb{Z}_q^n$ or generated uniformly over $\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$.
- Take some of them to the oracle of SLWE problem to find $\boldsymbol{s} \in \mathbb{Z}_q^n$.
- Then ?

"≥"

- Given $poly(n)$ samples a LWE distribution $\mathcal{A}_{\boldsymbol{s},n,q,\chi}$ for some unknown $\boldsymbol{s} \in \mathbb{Z}_q^n$. Claim we can solve it with a DLWE oracle.

If the modulus q is $poly(n)$-bounded, then DLWE=SLWE problem

"$\geq$"

- Given $poly(n)$ samples $(\boldsymbol{a_i}, b_i)$ from a LWE distribution $\mathcal{A}_{\boldsymbol{s},n,q,\chi}$ for some unknown $\boldsymbol{s} = (s_1, \dots, s_n)^T \in \mathbb{Z}_q^n$. Claim we can solve for s_1 with a DLWE oracle.

1. Choose guessing $k \in \mathbb{Z}_q$.
2. Define a transformation $\phi_k: \mathbb{Z}_q^n \times \mathbb{Z}_q \to \mathbb{Z}_q^n \times \mathbb{Z}_q$ by

$$(\boldsymbol{a} + r',\ b + k \cdot r) \leftarrow \phi_k(\boldsymbol{a}, b),$$

where $r \in \mathbb{Z}_q$ is generated uniformly at random and $r' = (r, 0, \dots, 0)^T \in \mathbb{Z}_q^n$.

<Uniformly random>:

$\Rightarrow$?

If the modulus q is $poly(n)$-bounded, then DLWE=SLWE problem

"$\geq$"

- Define a transformation $\phi_k: \mathbb{Z}_q^n \times \mathbb{Z}_q \to \mathbb{Z}_q^n \times \mathbb{Z}_q$ by

$$(\boldsymbol{a} + r',\ b + k \cdot r) \leftarrow \phi_k(\boldsymbol{a}, b),$$

where $r \in \mathbb{Z}_q$ is generated uniformly at random and $r' = (r, 0, \dots, 0)^T \in \mathbb{Z}_q^n$.

<LWE samples>: $\big((\boldsymbol{a_i}, b_i) = (\boldsymbol{a_i}, \langle \boldsymbol{a_i}, \boldsymbol{s} \rangle + e_i)\big)$

$$(\boldsymbol{a_i} + r', \qquad b_i + k \cdot r) = \big(\boxed{\boldsymbol{a_i} + \boldsymbol{r'}}, \qquad \langle \boxed{\boldsymbol{a_i} + \boldsymbol{r'}}, s \rangle + e_i + \underline{r \cdot (k - s_1)}\big)$$

If $k = s_1$, $\Rightarrow$?

If $k \neq s_1$, $\Rightarrow$?

Short Secret DLWE Problem:

Dimension n
Modulus q
Error distribution χ

Adversary

$(A, \boldsymbol{b})$ $\longleftarrow$

Challenger

Output: {"LWE", "uniform at random"}

$\boldsymbol{s} \leftarrow \chi^{\boldsymbol{n}}$

Generate :
$poly(n)$ LWE samples $(A, \boldsymbol{b})$
either from $\mathcal{A}_{\boldsymbol{s},n,q,\chi}$ or
uniformly at random over
$\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$

Lemma:
Short Secret DLWE Problem≥DLWE problem

- Given access to the short secret LWE problem.

- Given LWE instances ($A \in \mathbb{Z}_q^{n\times n}, \boldsymbol{b} = A^T s + \boldsymbol{e_s}$) where $A \xleftarrow{\$} \mathbb{Z}_q^{n\times n}$, unknown $s \xleftarrow{\$} \mathbb{Z}_q^n$ and e_s generated from χ. (DLWE problem setting)
- Say A is invertible.
- Consider a transformation $\phi: \mathbb{Z}_q^n \times \mathbb{Z}_q \rightarrow \mathbb{Z}_q^n \times \mathbb{Z}_q$
$$\phi(\boldsymbol{a}', b') = (-A^{-1}\boldsymbol{a}', b' + \langle -A^{-1}\boldsymbol{a}', b\rangle)$$
- Given a LWE instance ($\boldsymbol{a}' \in \mathbb{Z}_q^n, b' = \langle \boldsymbol{a}', s\rangle + e'$)
- Compute:$\phi(\boldsymbol{a}', b')$
$$= (-A^{-1}\boldsymbol{a}', \langle -A^{-1}\boldsymbol{a}', e_s\rangle + e')$$

$\Rightarrow$?

A Taste of
Passive Security Proof with DLWE problem

$$A \xleftarrow{\$} \mathbb{Z}_q^{n\times n}$$

$$\boldsymbol{s_a} \leftarrow \chi^n;\ \boldsymbol{e_a} \leftarrow \chi^n$$

$$\boldsymbol{b} = A\boldsymbol{s_a} + \boldsymbol{e_a} \in \mathbb{Z}_q^n$$

$$(A, \boldsymbol{b}) \in \mathbb{Z}_q^{n\times n} \times \mathbb{Z}_q^n \longrightarrow$$

$$\longleftarrow (\boldsymbol{b}', c) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$$

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

$$\boldsymbol{s_B}, \boldsymbol{e_B} \leftarrow \chi^n$$

$$e_B{}' \leftarrow \chi$$

$$\boldsymbol{b}' = \boldsymbol{s_B^T}A + \boldsymbol{e_B^T} \in \mathbb{Z}_q^n$$

$$c = \boldsymbol{s_B^T}\boldsymbol{b} + e_B' + m \cdot \left\lfloor \frac{q}{2} \right\rfloor$$

Decrypt:

$$c - \boldsymbol{b}'\boldsymbol{s_a} = e_b' - \boldsymbol{e_B^T}s_a \approx m \cdot \left\lfloor \frac{q}{2} \right\rceil (mod\ q)$$

Game0 (original protocol)

$A \overset{\$}{\leftarrow} \mathbb{Z}_q^{n\times n}$

$\boldsymbol{s_a} \leftarrow \chi^n;\ \boldsymbol{e_a} \leftarrow \chi^n$

$\boldsymbol{b} = A\boldsymbol{s_a} + \boldsymbol{e_a} \in \mathbb{Z}_q^n$

$(A, \boldsymbol{b})$ $\longrightarrow$

$(\boldsymbol{b}', c)$ $\longleftarrow$

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

$\boldsymbol{s_B}, \boldsymbol{e_B} \leftarrow \chi^n$

$e_B{}' \leftarrow \chi$

$\boldsymbol{b}' = \boldsymbol{s_B^T}A + \boldsymbol{e_B^T} \in \mathbb{Z}_q^n$

$c = \boldsymbol{s_B^T}\boldsymbol{b} + e_B' + m \cdot \left\lfloor \frac{q}{2} \right\rfloor$

Decrypt:

$c - \boldsymbol{b}'\boldsymbol{s_a} = e_B' - \boldsymbol{e_B^T}s_a$

$\approx m \cdot \left\lfloor \frac{q}{2} \right\rceil \pmod q$

$\boxed{\begin{array}{c}(A, \boldsymbol{b}) \\ (\boldsymbol{b}', c) \\ m\end{array}}$

Game1

$A \overset{\$}{\leftarrow} \mathbb{Z}_q^{n\times n}$

$\boldsymbol{b} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n}$

$(A, \boldsymbol{b}) \longrightarrow$

$\longleftarrow (\boldsymbol{b}', c)$

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

$\boldsymbol{s_B}, \boldsymbol{e_B} \leftarrow \chi^n$

$e_B{}' \leftarrow \chi$

$\boldsymbol{b}' = \boldsymbol{s_B^T} A + \boldsymbol{e_B^T} \in \mathbb{Z}_q^n$

$c = \boldsymbol{s_B^T}\boldsymbol{b} + e_B' + m \cdot \left\lfloor \frac{q}{2} \right\rfloor$

$$\boxed{\begin{array}{l}(A, \boldsymbol{b}) \\ (\boldsymbol{b}', c) \\ m\end{array}}$$

Game2

$A \xleftarrow{\$} \mathbb{Z}_q^{n\times n}$

$\boldsymbol{b} \xleftarrow{\$} \mathbb{Z}_q^{n}$

$(A, \boldsymbol{b}) \longrightarrow$

$(\boldsymbol{b}', c) \longleftarrow$

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

$\boldsymbol{s_B} \leftarrow \chi^n$

$e_B{}' \leftarrow \chi$

$\boldsymbol{b}' \xleftarrow{\$} \mathbb{Z}_q^{n}$

$c = \boldsymbol{s_B^T}\boldsymbol{b} + e_B' + m \cdot \left\lfloor \frac{q}{2} \right\rceil$

$$\boxed{\begin{array}{c} (A, \boldsymbol{b}) \\ (\boldsymbol{b}', c) \\ m \end{array}}$$

Game3

$A \xleftarrow{\$} \mathbb{Z}_q^{n\times n}$

$\boldsymbol{b} \xleftarrow{\$} \mathbb{Z}_q^{n}$

$(A, \boldsymbol{b})$

$(\boldsymbol{b}', c)$

$m \xleftarrow{\$} \{0,1\}$

$\boldsymbol{b}' \xleftarrow{\$} \mathbb{Z}_q^{n}$

$c \xleftarrow{\$} \mathbb{Z}_q$

$(A, \boldsymbol{b})$

$(\boldsymbol{b}', c)$

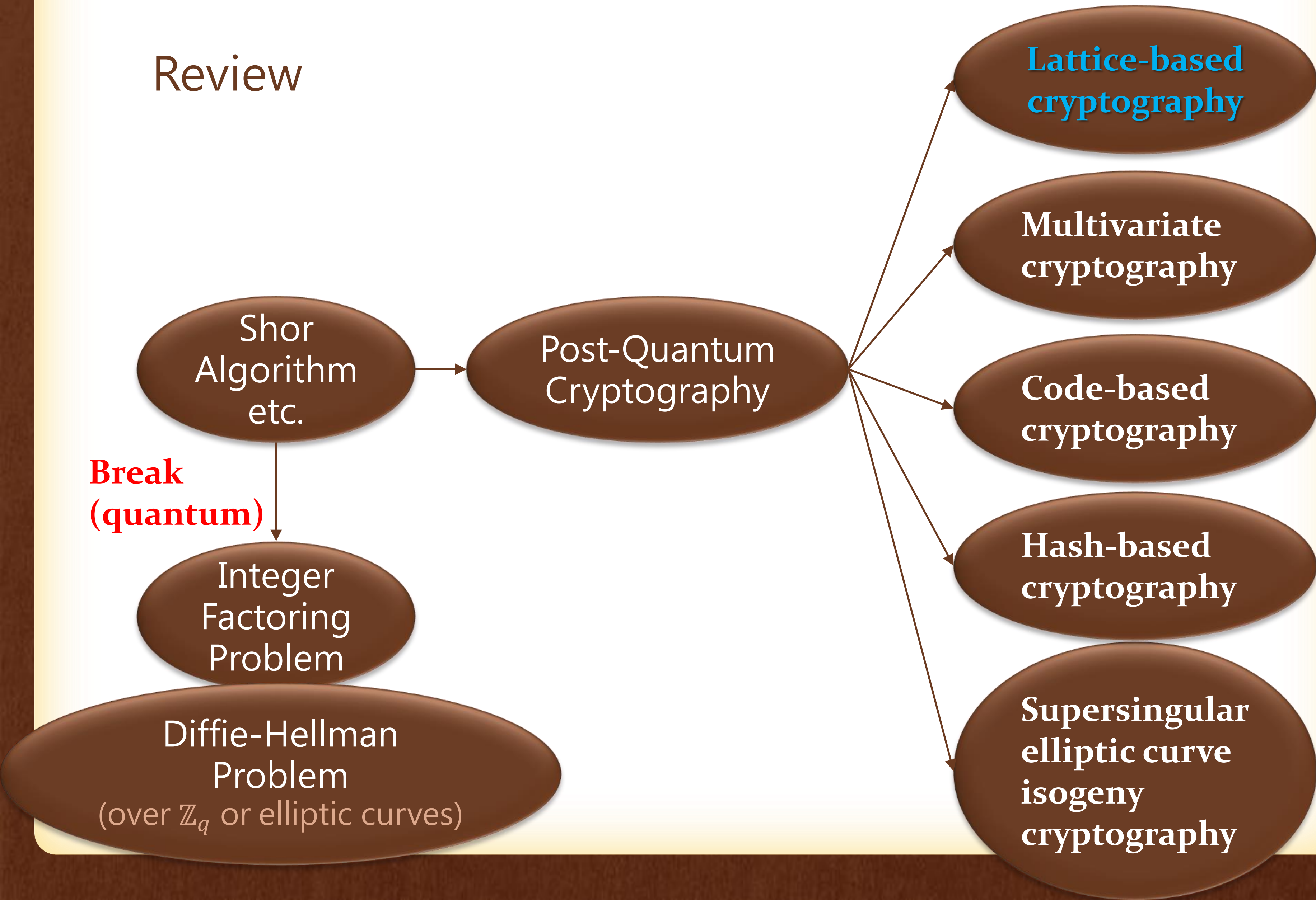
m

Public Key Cryptosystems Based on the LWE Problem or Related Problems

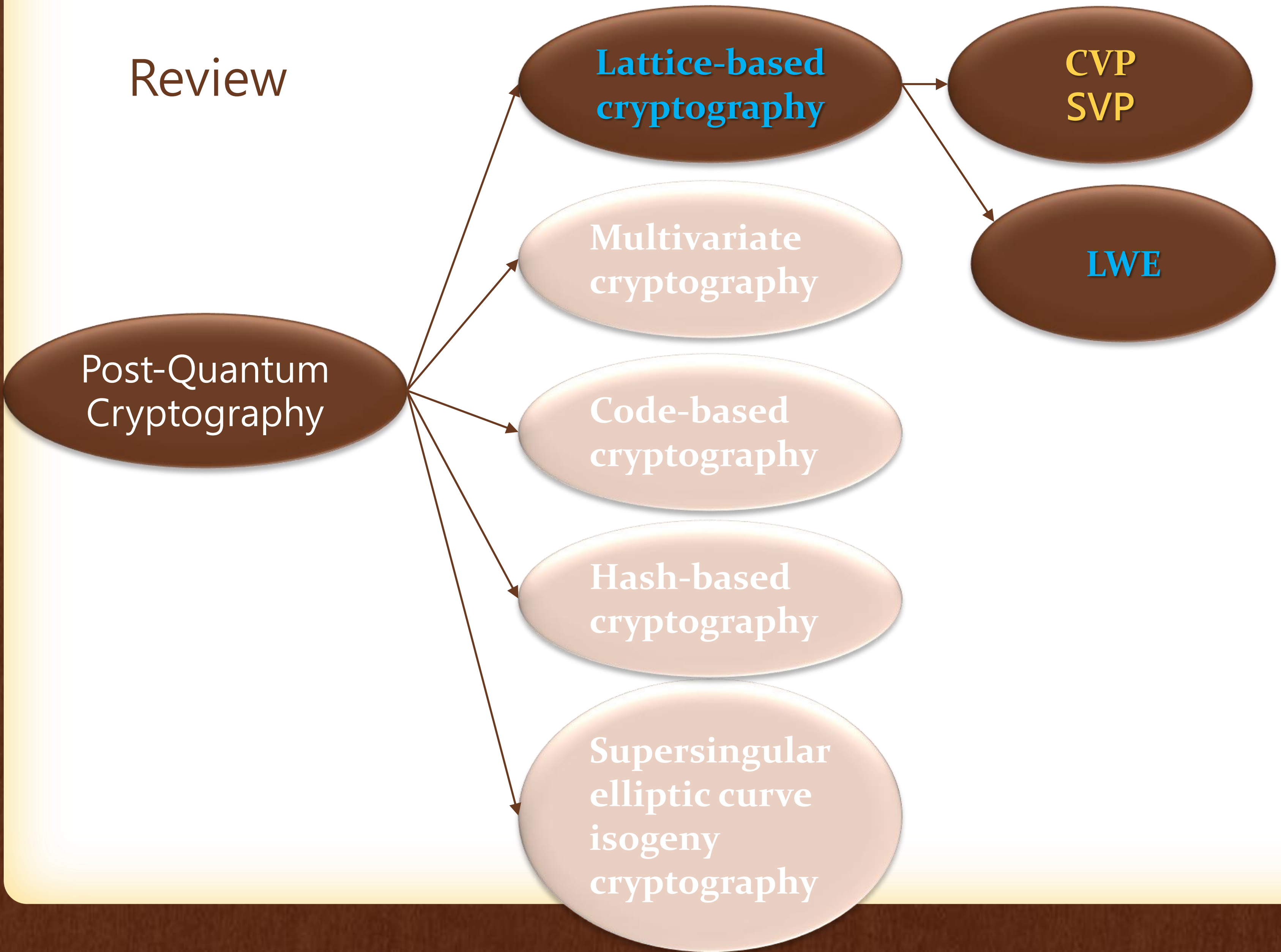
Contents

- Review
- Diffie-Hellman Like Key Exchange
- Peikert's Method
- RLWE in Brief
- NewHope

Review

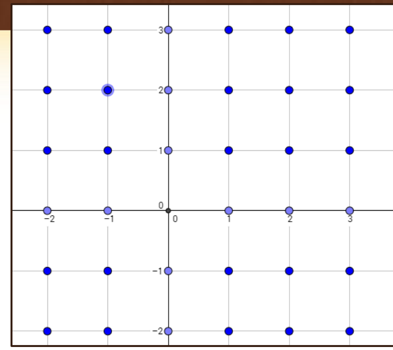

Shor Algorithm etc.
Post-Quantum Cryptography
Break (quantum)
Integer Factoring Problem
Diffie-Hellman Problem
(over $\mathbb{Z}_q$ or elliptic curves)
Lattice-based cryptography
Multivariate cryptography
Code-based cryptography
Hash-based cryptography
Supersingular elliptic curve isogeny cryptography

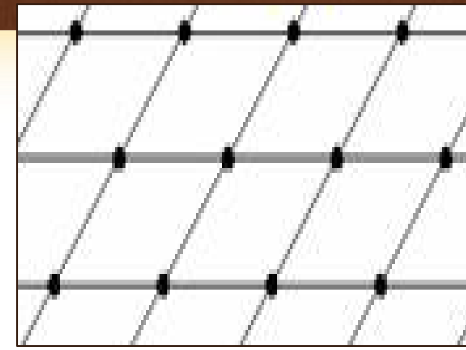
Review



Post-Quantum Cryptography
Lattice-based cryptography
CVP
SVP
LWE
Multivariate cryptography
Code-based cryptography
Hash-based cryptography
Supersingular elliptic curve isogeny cryptography

Lattices in Cryptography





- **Definition 1.** For a linear independent set $B = \{u_1, \dots, u_k\} \subset \mathbb{R}^n$, a lattice L generated by B in $\mathbb{R}^n$ is defined to be

$$L = \sum \mathbb{Z}u_i \,.$$

- Given a generating set of a lattice L in $\mathbb{R}^n$.
- ***Closest vector problem*** (CVP):
  Given: and a vector w in $\mathbb{R}^n$
  Request: Find $v \in L$ such that

$$\|w - v\| = \min_{u \in L} \|w - u\|$$

- ***Shortest vector problem*** (SVP):

  Request: Find a nonzero vector $v \in L$ such that

$$\|v\| = \min_{u \in L - \{0\}} \|u\|$$

Hardness

- Oded Regev shows that $LWE \geq approx{-}S(I)VP$ & $approx{-}GapSVP$ with a quantum reduction. There is also a classical one in a bit more wider condition provided by Chris Peikert.
- Specifically, with $dimension\ n$, modulus $q = poly(n)$, and error distribution (discrete normal/Gaussian distribution) χ of standard deviation αq,

$$LWE_{n,q,\chi} \geq (n/\alpha){-}S(I)VP.$$

Diffie-Hellman Like Structure [DXL12]

$$A \xleftarrow{\$} \mathbb{Z}_q^{n\times n}$$

$$\boldsymbol{s_a} \leftarrow \chi^n;\ \boldsymbol{e_a} \leftarrow \chi^n$$
$$\boldsymbol{b} = A\boldsymbol{s_a} + \boldsymbol{e_a} \in \mathbb{Z}_q^n$$

$$\boldsymbol{s_B}, \boldsymbol{e_B} \leftarrow \chi^n$$
$$\boldsymbol{b'} = A^T \boldsymbol{s_B} + \boldsymbol{e_B} \in \mathbb{Z}_q^n$$

$$(A, \boldsymbol{b}) \in \mathbb{Z}_q^{n\times n} \times \mathbb{Z}_q^n \longrightarrow$$

$$(\boldsymbol{b'}) \in \mathbb{Z}_q \longleftarrow$$

$$\boldsymbol{s_a^T b'} = \boxed{\boldsymbol{s_a^T} A^T \boldsymbol{s_B}} + \boldsymbol{s_a^T e_B} \ (mod\ q)$$

$$\boldsymbol{s_B^T b} = \boxed{\boldsymbol{s_B^T} A \boldsymbol{s_a}} + \boldsymbol{s_B^T e_a} \ (mod\ q)$$

Example:
Diffie-Hellman Like Structure

$$A = \begin{pmatrix} 2 & 42 & -31 \\ 45 & -26 & 21 \\ 14 & -5 & -23 \end{pmatrix}$$

$$\boldsymbol{s_a} = \begin{pmatrix} -1 \\ 2 \\ -2 \end{pmatrix};\ \boldsymbol{e_a} = \begin{pmatrix} 1 \\ 0 \\ -3 \end{pmatrix}$$

$$\boldsymbol{b} = A\boldsymbol{s_a} + \boldsymbol{e_a} = \begin{pmatrix} 40 \\ -24 \\ -37 \end{pmatrix}$$

$$\boldsymbol{s_B} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix};\ \boldsymbol{e_B} = \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix}$$

$$\boldsymbol{b'} = A^T\boldsymbol{s_B} + \boldsymbol{e_B} = \begin{pmatrix} -8 \\ -7 \\ 19 \end{pmatrix}$$

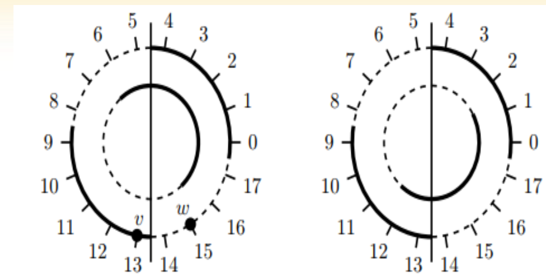
$$\xrightarrow{(A, \boldsymbol{b})}$$

$$\xleftarrow{(\boldsymbol{b'})}$$

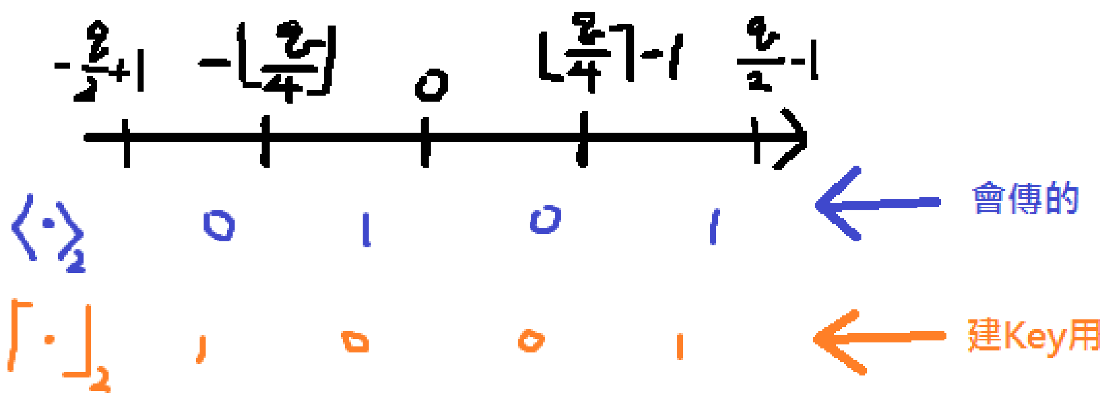
$$\boldsymbol{s_a^T b'} = \boldsymbol{s_a^T} A^T \boldsymbol{s_B} + \boldsymbol{s_a^T e_B} = 48\ (mod\ q)$$

$$\boldsymbol{s_B^T b} = \boldsymbol{s_B^T} A \boldsymbol{s_a} + \boldsymbol{s_B^T e_a} = -47\ (mod\ q)$$

Peikert's Key Exchange (Sketch)



Chris Peikert



Error Tolerance: $\|s_a e_b - s_b e_a\|_\infty \le q/8$

Example:
Peikert's Method

$$A = \begin{pmatrix} 2 & 42 & -31 \\ 45 & -26 & 21 \\ 14 & -5 & -23 \end{pmatrix}$$

$$s_a = \begin{pmatrix} -1 \\ 2 \\ -2 \end{pmatrix};\ e_a = \begin{pmatrix} 1 \\ 0 \\ -3 \end{pmatrix}$$

$$b = As_a + e_a = \begin{pmatrix} 40 \\ -24 \\ -37 \end{pmatrix}$$

$$s_B = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix};\ e_B = \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix}$$

$$b' = A^T s_B + e_B = \begin{pmatrix} -8 \\ -7 \\ 19 \end{pmatrix}$$

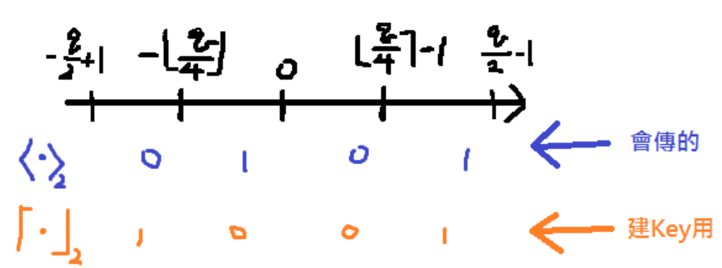
$$s_a^T b' = 48\ (mod\ q)$$

$$key = 1$$

$$\langle s_B^T b \rangle_2 = 0$$

$$s_B^T b = -47\ (mod\ q)$$

$$key = \lfloor s_B^T b \rceil_2 = 1$$



Remark.

- The previous scheme has been (over) simplified.

- Is the output, conditioned on the transmitting messages, unbiased?

Drawback:

1. Public key size (bit length) $\approx n^2 \cdot \log_2 q$

2. Transmission bandwidth $\approx 2 \cdot (n \cdot \log_2 q)$

3. Computation $\approx 2n^2$ (modular multiplications)

Intuition of Ring-LWE (on the public key).

- In LWE problem (or LWE-based cryptosystem), we need a matrix $A \overset{\$}{\leftarrow} \mathbb{Z}_q^{n \times n}$ or $A \overset{\$}{\leftarrow} \mathbb{Z}_q^{m \times n}$.

$$
\begin{array}{|c|}
\hline
\boldsymbol{a}_1 \\
\hline
\boldsymbol{a}_2 \\
\hline
\boldsymbol{a}_3 \\
\hline
\vdots \\
\hline
\\
\hline
\end{array}
$$

- What if we only generated the first row uniformly at random and generating an anti-cyclic matrix ?

$$
\begin{array}{|c|}
\hline
(\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_n) \\
\hline
(-\boldsymbol{c}_n, \boldsymbol{c}_1, \dots, \boldsymbol{c}_{n-1}) \\
\hline
(-\boldsymbol{c}_{n-1}, -\boldsymbol{c}_n, \dots, \boldsymbol{c}_{n-2}) \\
\hline
\vdots \\
\hline
(-\boldsymbol{c}_{n-2}, -\boldsymbol{c}_{n-1}, \dots, \boldsymbol{c}_{n-3}) \\
\hline
\end{array}
$$

- (The multiplication can be sped up in some specific condition.)

- Hardness Guarantee?

Hardness of Ring LWE

- Ring-LWE>approx-SIVP over ideal lattices.

- **Suspicion:**
  1. **Gap-SVP is not hard on ideal lattices.**
  2. **SVP/SIVP is an NP-hard problem.**
     SIVP over ideal lattices???

-

On Ideal Lattices and
Learning with Errors Over Rings*

Vadim Lyubashevsky† Chris Peikert‡ Oded Regev§

April 24, 2012

Abstract

rning with errors" (LWE) problem is to distinguish random linear equations, whi
y a small amount of noise, from truly uniform ones. The problem has been sh
st-case lattice problems, and in recent years it has served as the foundation f
phic applications. Unfortunately, these applications are rather inefficient due t
verhead in the use of LWE. A main open question was whether LWE and its
de truly efficient by exploiting extra algebraic structure, as was done for lattic
nd related primitives).
lve this question in the affirmative by introducing an algebraic variant of LWE

Ring-LWE (RLWE)

- LWE is over $\mathbb{Z}_q^{m \times n}$.
- Ring-LWE is over a ring R.
- Consider an ideal lattice on it.
- Typically, $R = \mathbb{Z}[x]/\langle x^n + 1 \rangle$ where n is power-of-2.

Omitting the tough version, we introduce a simple and widely used one.

Contents

- Why Do We Need RLWE?
- Some Math (no proofs)
  - Algebraic Integers and Ring of Integers
  - Canonical Mapping and Norm
  - Dedekind Domain
  - Trace and Dual
  - Different and Codifferent
- Problems on Ideal Lattices

Ring-DLWE Problem

- $R = \mathbb{Z}[x]/< x^n + 1 >$, where n is power–of–2
- $R_q = R/qR$, where $q \in \mathbb{N}$
- $\mathcal{X}$: an error distribution over R

- For $s \stackrel{\$}{\leftarrow} \chi$, the RLWE distribution $A_{s,\mathcal{X},q,n}$ over $R_q \times R_q$ which is sampled by choosing $a \in R_q$ uniformly at random and $e \stackrel{\$}{\leftarrow} \mathcal{X}$, and output $\big(a, b = (a \cdot s + e \bmod q)\big)$
- Given poly(n) samples from $A_{s,\mathcal{X},q,n}$ or uniformly random in $R_q \times R_q$, determine where the samples from.

Diffie-Hellman Like Structure (Ring LWE form)

$$R \;= \mathbb{Z}[x]/< x^n + 1 >$$
$$R_q = R/qR$$
$\mathcal{X}$: an error distribution over R

$A \overset{\$}{\leftarrow} R_q$

$s_a, e_a \overset{\$}{\leftarrow} \chi$

$u_a = As_a + e_a$

$(A, u_a = As_a + e_a)$ $\longrightarrow$

$\longleftarrow$ $u_b = As_b + e_b$

$s_b, e_b \overset{\$}{\leftarrow} \chi$

$u_b = As_b + e_b$

$$v_a = s_a u_b$$
$$= s_a A s_b + s_a e_b$$

$$\boxed{v_a - v_b = s_a e_b - s_b e_a}$$

$$v_b = s_b u_a$$
$$= s_b A s_a + s_b e_a$$

If $\mathcal{X}$ is β-bounded, then
$\|v_a - v_b\|_\infty = \|s_a e_b - s_b e_a\|_\infty \leq 2n\beta^2$
with overwhelming probability

Google Security Blog

The latest news and insights from Google on security and safety on the Internet

Expe

July 7,

Posted b

We're indebted to Erdem Alkim, Léo Ducas, Thomas Pöppelmann and Peter Schwabe, the researchers who developed "New Hope", the post-quantum algorithm that we selected for this experiment. Their scheme looked to be the most promising post-quantum key-exchange when we investigated in December 2015. Their work builds upon e

Lyuba

This experiment is currently enabled in Chrome Canary and you can tell whether it's being used by opening the recently introduced Security Panel and looking for "CECPQ1", for example on https://play.google.com/store. Not all Google domains will have it enabled and the experiment may appear and disappear a few times if any issues are found.

走在網路科技最前線

Google Chrome Canary 支援最新版 Chrome 的全新功能。
預先警告：這是專為開發人員和率先測試者設計的版本，有時可能會造成瀏覽器全面當機。

下載 Chrome Canary

膽小者請勿輕易嘗試

Canary 是專為開發人員和率先測試者設計的版本，可能會造成瀏覽器當機。

每晚更新

Canary 幾乎每天都會進行變更並安裝新功能。

並行安裝

為協助進行開發及測試，Canary 和 **Chrome 穩定版**可同時執行。

ImperialViolet

CECPQ1 results (28 Nov 2016)

Here the results are more concrete: we did not find any unexpected impediment to deploying something like NewHope. There were no reported problems caused by enabling it.

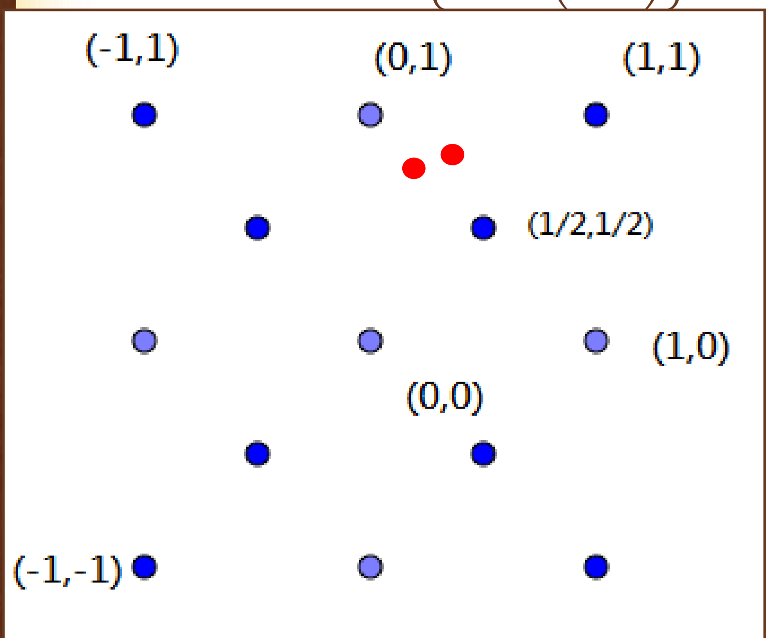
Although the median connection latency only increased by a millisecond, the latency for the slowest 5% increased by 20ms and, for the slowest 1%, by 150ms. Since NewHope is computationally inexpensive, we're assuming that this is caused entirely by the increased message sizes. Since connection latencies compound on the web (because subresource discovery is delayed), the data requirement of NewHope is moderately expensive for people on slower connections.

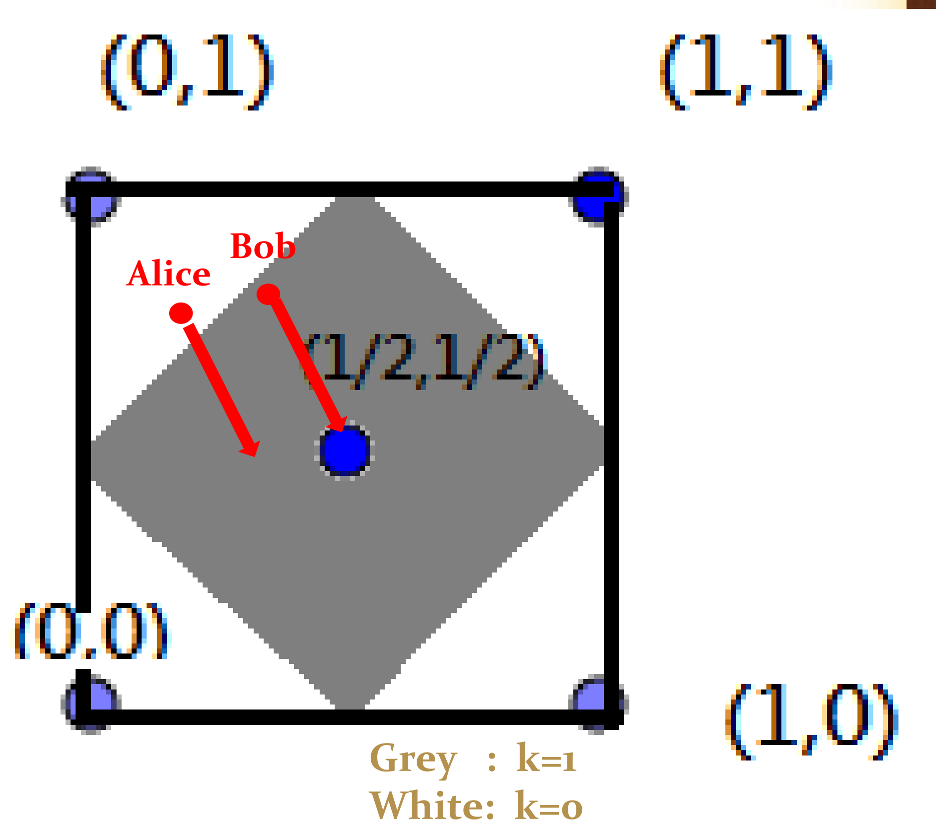
- The following is a “sketch” of the reconciliation mechanism.

Reconciliation in New Hope – Intuition1 (2 in 1 out)

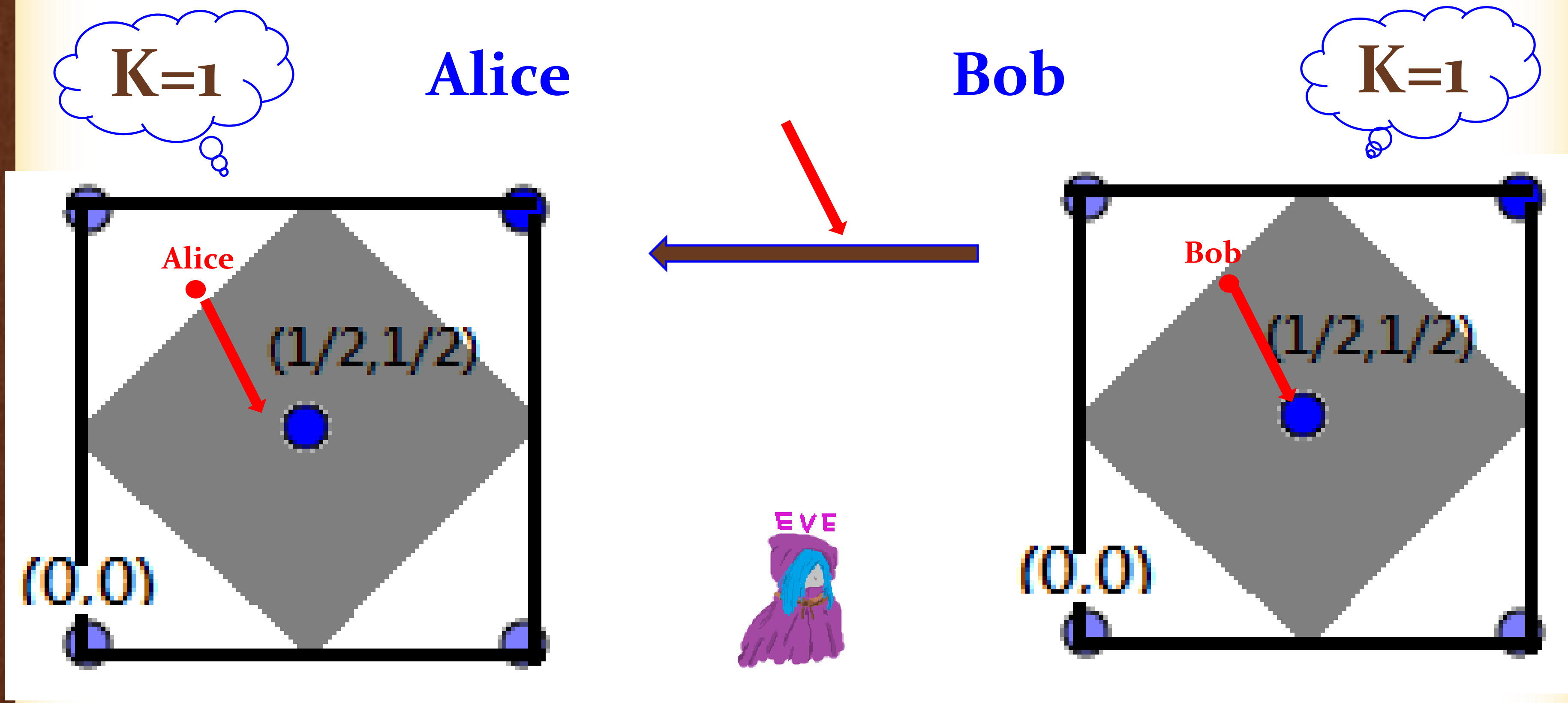
- $(v_0, v_1) \in \mathbb{Z}_q^2$, consider $(v_0/q, v_1/q) \in \mathbb{R}^2$

Lattice generated by $\left\{(1,0), \left(\frac{1}{2}, \frac{1}{2}\right)\right\}$

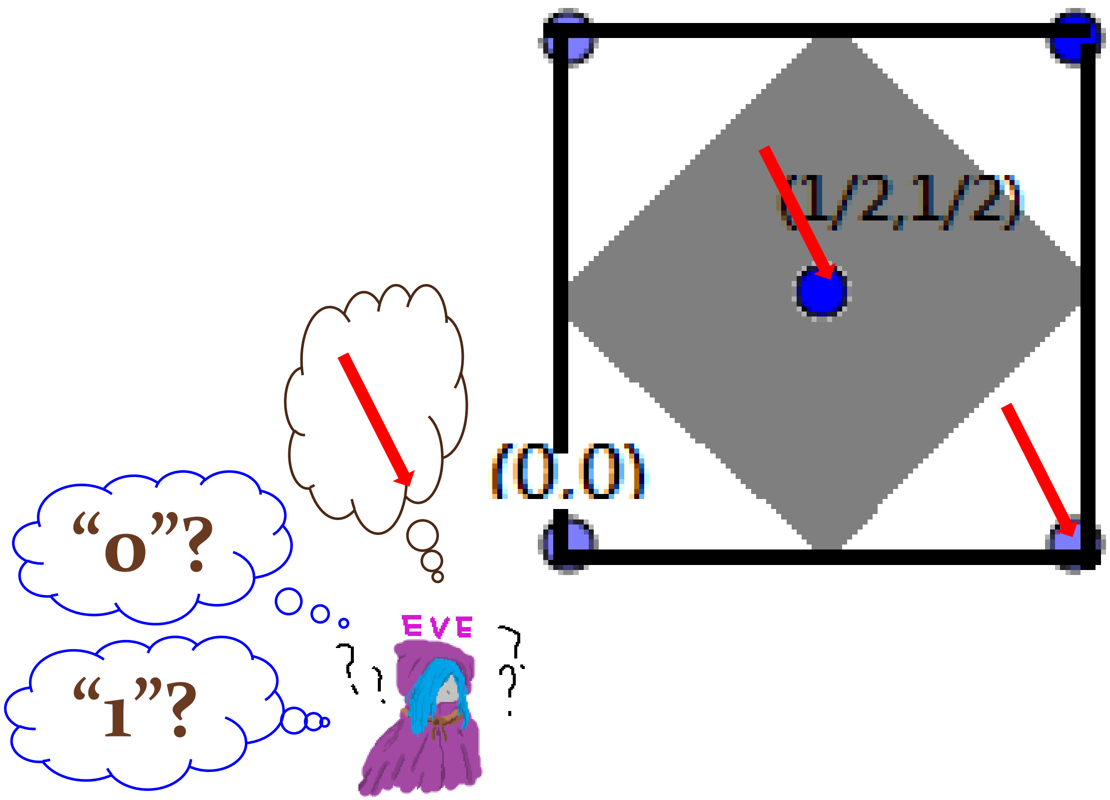




Example

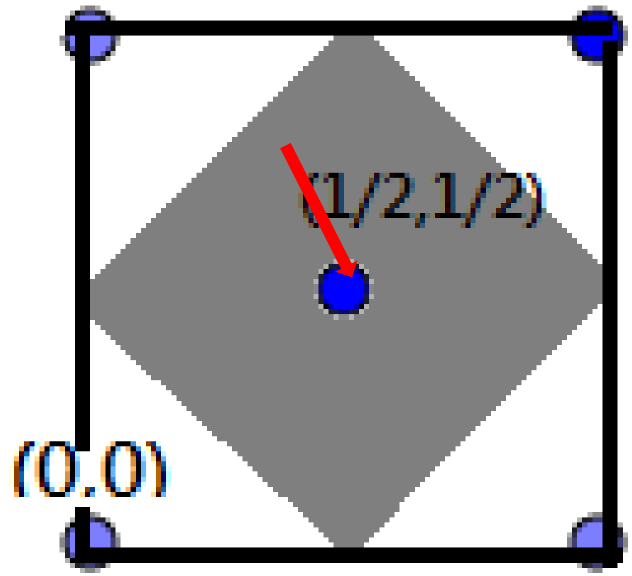


Reconciliation in New Hope - Security

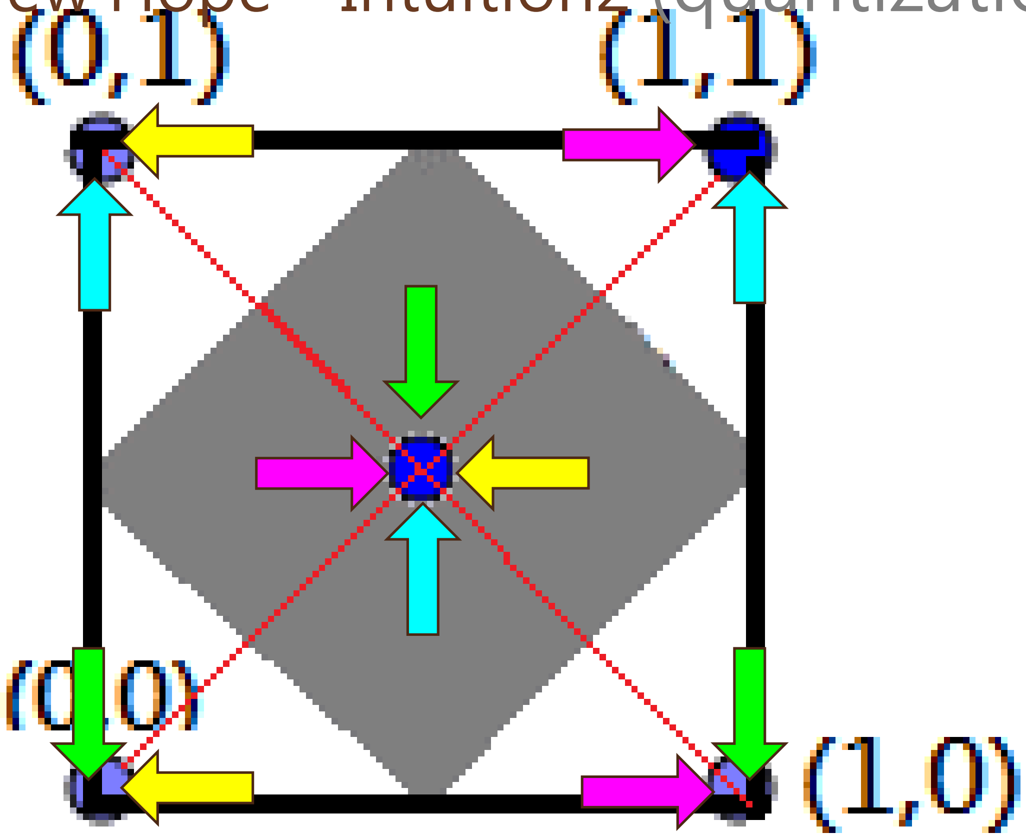


Drawback

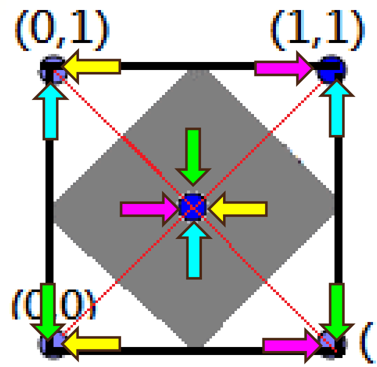
- Whole “difference vector” is too heavy

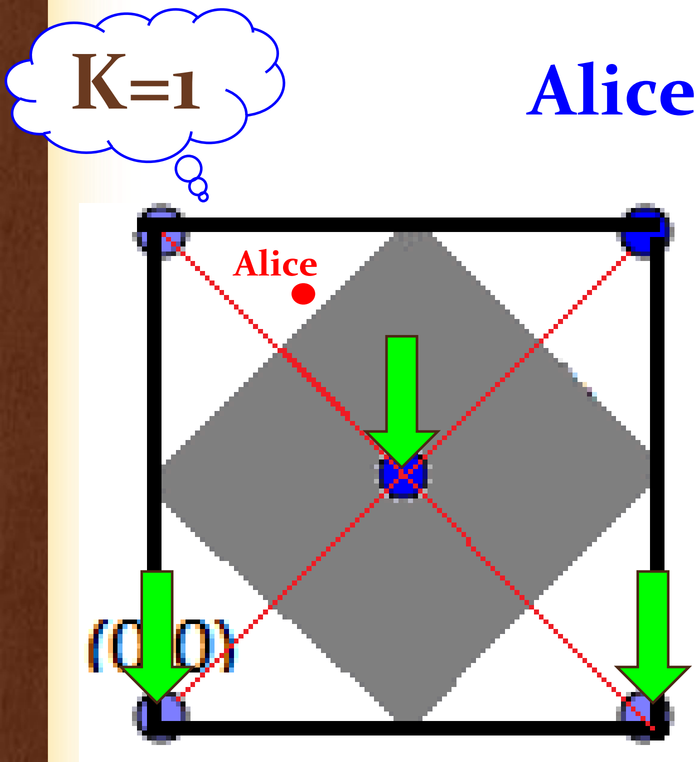


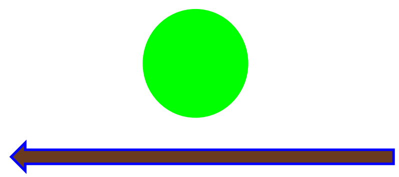
Reconciliation in New Hope – Intuition2 (quantization)

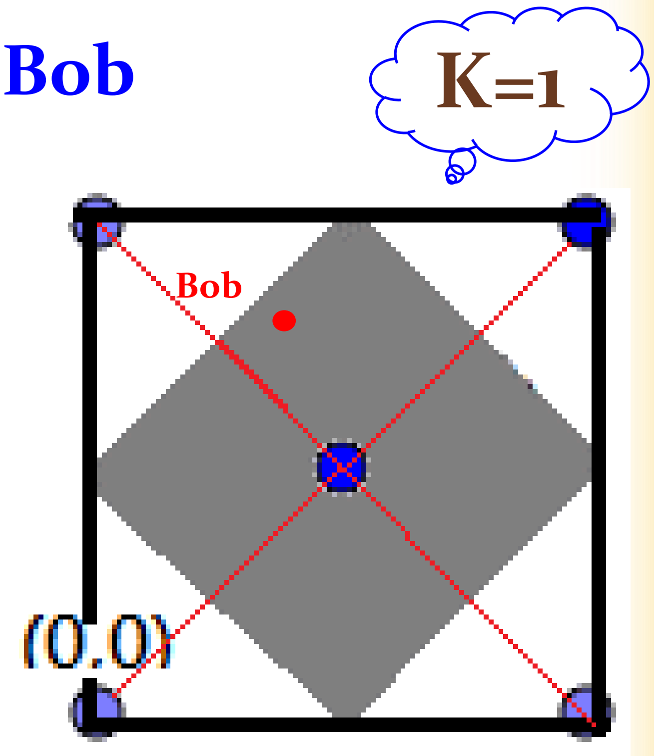


Example









Reconciliation in New Hope

- $\widetilde{D_4} \subseteq \mathbb{Z}^4$ *be a full lattice with basis* $\left\{u_1, u_2, u_3, g = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)\right\}$,

  where $\{u_i\}_{i=1}^4$ *is the standard basis for* $\mathbb{Z}^4$
- $\boldsymbol{CVP}_{\widetilde{\boldsymbol{D_4}}}$: *an CVP algorithm from* $\mathbb{R}^4$ *to* $\widetilde{D_4}$
- $\boldsymbol{HelpRec}\left(\boldsymbol{x}; \boldsymbol{b} \xleftarrow{\$} \{\boldsymbol{0}, \boldsymbol{1}\}\right)$: *an alg. encoding the location information*
- $\boldsymbol{Rec}(\boldsymbol{x}, \boldsymbol{r})$: *reconciliation with the information r*

Note: Including b, there are some technique in *HelpRec* to remove the bias.

New Hope

Parameters: $q = 12289 < 2^{14}$, $n = 1024$

Error distribution: ψ_{16}

Alice (server)		**Bob (client)**
$seed \xleftarrow{\$} \{0,1\}^{256}$		
$\mathbf{a} \leftarrow \mathsf{Parse}(\mathsf{SHAKE\text{-}128}(seed))$		
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \psi_{16}^{n}$		$\mathbf{s}', \mathbf{e}', \mathbf{e}'' \xleftarrow{\$} \psi_{16}^{n}$
$\mathbf{b} \leftarrow \mathbf{as} + \mathbf{e}$	$\xrightarrow{(\mathbf{b}, seed)}$	$\mathbf{a} \leftarrow \mathsf{Parse}(\mathsf{SHAKE\text{-}128}(seed))$
		$\mathbf{u} \leftarrow \mathbf{as}' + \mathbf{e}'$
		$\mathbf{v} \leftarrow \mathbf{bs}' + \mathbf{e}''$
$\mathbf{v}' \leftarrow \mathbf{us}$	$\xleftarrow{(\mathbf{u}, \mathbf{r})}$	$\mathbf{r} \xleftarrow{\$} \mathsf{HelpRec}(\mathbf{v})$
$\nu \leftarrow \mathsf{Rec}(\mathbf{v}', \mathbf{r})$		$\nu \leftarrow \mathsf{Rec}(\mathbf{v}, \mathbf{r})$
$\mu \leftarrow \mathsf{SHA3\text{-}256}(\nu)$		$\mu \leftarrow \mathsf{SHA3\text{-}256}(\nu)$

From the paper

ψ_{16}

Parameters: $q = 12289 < 2^{14}$, $n = 1024$

Error distribution: ψ_{16}

Alice (server)		**Bob (client)**
$seed \xleftarrow{\$} \{0,1\}^{256}$		
$\mathbf{a} \leftarrow \mathsf{Parse}(\mathsf{SHAKE\text{-}128}(seed))$		
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \psi_{16}^{n}$		$\mathbf{s}', \mathbf{e}', \mathbf{e}'' \xleftarrow{\$} \psi_{16}^{n}$
$\mathbf{b} \leftarrow \mathbf{a}\mathbf{s} + \mathbf{e}$	$\xrightarrow{(\mathbf{b}, seed)}$	$\mathbf{a} \leftarrow \mathsf{Parse}(\mathsf{SHAKE\text{-}128}(seed))$
		$\mathbf{u} \leftarrow \mathbf{a}\mathbf{s}' + \mathbf{e}'$
		$\mathbf{v} \leftarrow \mathbf{b}\mathbf{s}' + \mathbf{e}''$
$\mathbf{v}' \leftarrow \mathbf{u}\mathbf{s}$	$\xleftarrow{(\mathbf{u}, \mathbf{r})}$	$\mathbf{r} \xleftarrow{\$} \mathsf{HelpRec}(\mathbf{v})$
$\nu \leftarrow \mathsf{Rec}(\mathbf{v}', \mathbf{r})$		$\nu \leftarrow \mathsf{Rec}(\mathbf{v}, \mathbf{r})$
$\mu \leftarrow \mathsf{SHA3\text{-}256}(\nu)$		$\mu \leftarrow \mathsf{SHA3\text{-}256}(\nu)$

From the paper

ψ_{16} :

Replace Rounded Gaussian with binomial distribution

- High-precision Gaussian sampling is expensive
- ψ_{16} : the centered binomial distribution of variance 8 is rather trivial in hardware and software without decreasing too much security for the protocol
- More precisely,

Theorem 4.1 *Let ξ be the rounded Gaussian distribution of parameter $\sigma = \sqrt{8}$, that is, the distribution of $\lfloor \sqrt{8} \cdot x \rceil$ where x follows the standard normal distribution. Let $\mathscr{P}$ be the idealized version of Protocol 2, where the distribution ψ_{16} is replaced by ξ. If an (unbounded) algorithm, given as input the transcript of an instance of Protocol 2 succeeds in recovering the pre-hash key ν with probability p, then it would also succeed against $\mathscr{P}$ with probability at least*

$$q \geq p^{9/8}/26.$$

Fresh $\mathbf{a}$

Parameters: q, n, χ

KEM.Setup() :

$\mathbf{a} \xleftarrow{\$} \mathcal{R}_q$

Alice (server)		Bob (client)
KEM.Gen($\mathbf{a}$) :		KEM.Encaps($\mathbf{a}, \mathbf{b}$) :
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \chi$		$\mathbf{s}', \mathbf{e}', \mathbf{e}'' \xleftarrow{\$} \chi$
$\mathbf{b} \leftarrow \mathbf{a}\mathbf{s} + \mathbf{e}$	$\xrightarrow{\mathbf{b}}$	$\mathbf{u} \leftarrow \mathbf{a}\mathbf{s}' + \mathbf{e}'$
		$\mathbf{v} \leftarrow \mathbf{b}\mathbf{s}' + \mathbf{e}''$
		$\bar{\mathbf{v}} \xleftarrow{\$} \mathsf{dbl}(\mathbf{v})$
KEM.Decaps($\mathbf{s}, (\mathbf{u}, \mathbf{v}')$) :	$\xleftarrow{\mathbf{u}, \mathbf{v}'}$	$\mathbf{v}' = \langle \bar{\mathbf{v}} \rangle_2$
$\mu \leftarrow \mathsf{rec}(2\mathbf{u}\mathbf{s}, \mathbf{v}')$		$\mu \leftarrow \lfloor \bar{\mathbf{v}} \rceil_2$

Parameters: $q = 12289 < 2^{14}$, $n = 1024$

Error distribution: ψ_{16}

Alice (server)		
$seed \xleftarrow{\$} \{0,1\}^{256}$		
$\mathbf{a} \leftarrow \mathsf{Parse}(\mathsf{SHAKE\text{-}128}(seed))$		
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \psi_{16}^n$		$\mathbf{s}', \mathbf{e}', \mathbf{e}'' \xleftarrow{\$} \psi_{16}^n$
$\mathbf{b} \leftarrow \mathbf{a}\mathbf{s} + \mathbf{e}$	$\xrightarrow{(\mathbf{b}, seed)}$	$\mathbf{a} \leftarrow \mathsf{Parse}(\mathsf{SHAKE\text{-}128}(seed))$
		$\mathbf{u} \leftarrow \mathbf{a}\mathbf{s}' + \mathbf{e}'$
		$\mathbf{v} \leftarrow \mathbf{b}\mathbf{s}' + \mathbf{e}''$
$\mathbf{v}' \leftarrow \mathbf{u}\mathbf{s}$	$\xleftarrow{(\mathbf{u}, \mathbf{r})}$	$\mathbf{r} \xleftarrow{\$} \mathsf{HelpRec}(\mathbf{v})$
$\nu \leftarrow \mathsf{Rec}(\mathbf{v}', \mathbf{r})$		$\nu \leftarrow \mathsf{Rec}(\mathbf{v}, \mathbf{r})$
$\mu \leftarrow \mathsf{SHA3\text{-}256}(\nu)$		$\mu \leftarrow \mathsf{SHA3\text{-}256}(\nu)$

From the paper

Fresh $\boldsymbol{a}$:

Against the all-for-the-price-of-one attacks

Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice

David Adrian[¶] Karthikeyan Bhargavan[*] Zakir Durumeric[¶] Pierrick Gaudry[†] Matthew Green[§]
J. Alex Halderman[¶] Nadia Heninger[‡] Drew Springall[¶] Emmanuel Thomé[†] Luke Valenta[‡]
Benjamin VanderSloot[¶] Eric Wustrow[¶] Santiago Zanella-Béguelin[‖] Paul Zimmermann[†]

[*] INRIA Paris-Rocquencourt [†] INRIA Nancy-Grand Est, CNRS, and Université de Lorraine
[‖] Microsoft Research [‡] University of Pennsylvania [§] Johns Hopkins [¶] University of Michigan

For additional materials and contact information, visit WeakDH.org.

ABSTRACT

We investigate the security of Diffie-Hellman key exchange as used in popular Internet protocols and find it to be less secure than widely believed. First, we present Logjam, a novel flaw in TLS that lets a man-in-the-middle downgrade connections to "export-grade" Diffie-Hellman. To carry out this attack, we implement the number field sieve discrete log algorithm. After a week-long precomputation for a specified 512-bit

coded, or widely shared Diffie-Hellman parameters has the effect of dramatically reducing the cost of large-scale attacks, bringing some within range of feasibility today.

The current best technique for attacking Diffie-Hellman relies on compromising one of the private exponents (a, b) by computing the discrete log of the corresponding public value ($g^a \bmod p$, $g^b \bmod p$). With state-of-the-art number field sieve algorithms, computing a single discrete log is more difficult than factoring an RSA modulus of the same size

Unusually Large n=1024

Parameters: $q = 12289 < 2^{14}$, $n = 1024$
Error distribution: ψ_{16}

Alice (server)		**Bob (client)**
$seed \xleftarrow{\$} \{0,1\}^{256}$		
$\mathbf{a} \leftarrow \mathsf{Parse}(\mathsf{SHAKE\text{-}128}(seed))$		
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \psi_{16}^{n}$		$\mathbf{s}', \mathbf{e}', \mathbf{e}'' \xleftarrow{\$} \psi_{16}^{n}$
$\mathbf{b} \leftarrow \mathbf{as} + \mathbf{e}$	$\xrightarrow{(\mathbf{b}, seed)}$	$\mathbf{a} \leftarrow \mathsf{Parse}(\mathsf{SHAKE\text{-}128}(seed))$
		$\mathbf{u} \leftarrow \mathbf{as}' + \mathbf{e}'$
		$\mathbf{v} \leftarrow \mathbf{bs}' + \mathbf{e}''$
$\mathbf{v}' \leftarrow \mathbf{us}$	$\xleftarrow{(\mathbf{u}, \mathbf{r})}$	$\mathbf{r} \xleftarrow{\$} \mathsf{HelpRec}(\mathbf{v})$
$\nu \leftarrow \mathsf{Rec}(\mathbf{v}', \mathbf{r})$		$\nu \leftarrow \mathsf{Rec}(\mathbf{v}, \mathbf{r})$
$\mu \leftarrow \mathsf{SHA3\text{-}256}(\nu)$		$\mu \leftarrow \mathsf{SHA3\text{-}256}(\nu)$

Unusually Large n=1024

- In security of level 128, the previous schemes are built with n at most 512
- The security of LWE-based schemes has not been thoroughly studied
- In view of RSA, the standardization and deployment of a scheme awakens further cryptanalytic effort

q=1 mod 2n and 14-bit Prime

Parameters: $q = 12289 < 2^{14}$, $n = 1024$

Error distribution: ψ_{16}

Alice (server)		**Bob (client)**
$seed \xleftarrow{\$} \{0,1\}^{256}$		
$\mathbf{a} \leftarrow \mathsf{Parse}(\mathsf{SHAKE\text{-}128}(seed))$		
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \psi_{16}^{n}$		$\mathbf{s}', \mathbf{e}', \mathbf{e}'' \xleftarrow{\$} \psi_{16}^{n}$
$\mathbf{b} \leftarrow \mathbf{a}\mathbf{s} + \mathbf{e}$	$\xrightarrow{(\mathbf{b}, seed)}$	$\mathbf{a} \leftarrow \mathsf{Parse}(\mathsf{SHAKE\text{-}128}(seed))$
		$\mathbf{u} \leftarrow \mathbf{a}\mathbf{s}' + \mathbf{e}'$
		$\mathbf{v} \leftarrow \mathbf{b}\mathbf{s}' + \mathbf{e}''$
$\mathbf{v}' \leftarrow \mathbf{u}\mathbf{s}$	$\xleftarrow{(\mathbf{u}, \mathbf{r})}$	$\mathbf{r} \xleftarrow{\$} \mathsf{HelpRec}(\mathbf{v})$
$\nu \leftarrow \mathsf{Rec}(\mathbf{v}', \mathbf{r})$		$\nu \leftarrow \mathsf{Rec}(\mathbf{v}, \mathbf{r})$
$\mu \leftarrow \mathsf{SHA3\text{-}256}(\nu)$		$\mu \leftarrow \mathsf{SHA3\text{-}256}(\nu)$

From the paper

Number-Theoretic Transform (NTT)

- Algorithm speeding up modular polynomial multiplications
- $a, b \in \mathcal{R}$. $Compute\ ab = NTT^{-1}\big(NTT(a) \cdot NTT(b)\big)$
- **q=1 mod 2n**: ensures the n^{th} root of unity ω & its square root γ exist in $\mathbb{Z}_q$, auxiliary elements in the algorithm.

e.g: For q=12287, n=1024 here, $\omega = 49$ and $\gamma = 7$

with $\omega^{1024} = 1\ mod\ q$ and $\omega = \gamma^2$

NTT

$$\mathsf{NTT}(\mathbf{g}) = \hat{\mathbf{g}} = \sum_{i=0}^{1023} \hat{g}_i X^i, \text{ with}$$

$$\hat{g}_i = \sum_{j=0}^{1023} \gamma^j g_j \omega^{ij},$$

$$\mathsf{NTT}^{-1}(\hat{\mathbf{g}}) = \mathbf{g} = \sum_{i=0}^{1023} g_i X^i, \text{ with}$$

$$g_i = n^{-1}\gamma^{-i} \sum_{j=0}^{1023} \hat{g}_j \omega^{-ij}.$$

- With the Montgomery multiplication constant $R = 2^{18} > q < 2^{14}$
- $2^{18}2^{14} \leq 2^{32}$ (within 32-bits) is nice for software implementation

Parameter Considerations in Conclusion

Speed/cost:

- $q = 1\ mod\ 2n$ & $q < 2^{14}$ (NTT)
- ψ_{16} (cost down)
- Improved Rec/HelpRec (decreasing the modulus q)

Security/failure:

- Fresh $\boldsymbol{a}$ (Against the all-for-the-price-of-one attacks)
- Improved Rec/HelpRec
- $n = 1024$ (be wary)

Reference

- Google Security Blog:
  https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html
- CECPQ1results:
  https://www.imperialviolet.org/2016/11/28/cecpq1.html

- V. Lyubashevsky, C. Peikert, and O. Regev. *On ideal lattices and learning with errors over rings.*
- E. Alkim, Léo Ducas, Thomas Pöppelmann, Peter Schwabe . *Post-quantum key exchange - a new hope.*
- Chris Peikert. Lattice cryptography for the internet. In International Workshop on Post-Quantum Cryptography, pages 197–219. Springer, 2014.
- Jintai Ding. A simple provably secure key exchange scheme based on the learning with errors problem. Cryptology ePrint Archive, Report 2012/688, 2012. https://eprint.iacr.org/2012/688/20121210:115748.
- Jintai Ding, Xiang Xie, and Xiaodong Lin. A simple provably secure key exchange scheme based on the learning with errors problem. Cryptology ePrint Archive, Report 2012/688 (29 Jul 2014 revised), 2012. http://eprint.iacr.org/2012/688.
- Oded Regev. The learning with errors problem. Invited survey in CCC, page 15, 2010.

Reference

- Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In Annual International Cryptology Conference, pages 554–571. Springer, 2008.
- Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. SIAM Journal on Computing, 40(6):1803–1844, 2011.
- Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In Proceedings of the forty-first annual ACM symposium on Theory of computing, pages 333–342. ACM, 2009.
- Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory (TOCT), 6(3):13, 2014.
- Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In Annual International Cryptology Conference, pages 554–571. Springer, 2008.
- Gerald J.Janusz. Algebraic Number Fields, 2^{nd} ed.

END