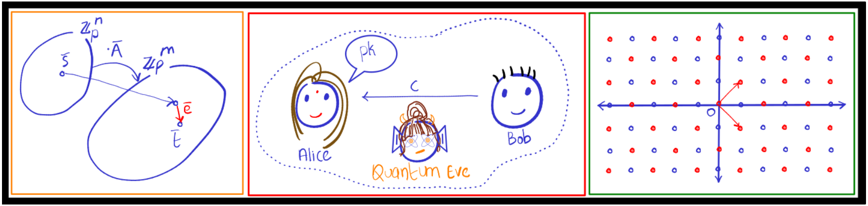


CS783: Theoretical Foundations of Cryptography

Lecture 10 (03/Sep/24)

Instructor: Chethan Kamath

Recall from Last Lecture

- Task 4: Public-key encryption
  - Modelled setting and security (CPA secrecy)

Recall from Last Lecture

- Task 4: Public-key encryption
  - Modelled setting and security (CPA secrecy)

  - Saw two CPA-secret constructions, with proofs:
    - ElGamal PKE ← DDH assumption ← $(g^a, g^b, g^{ab}) \approx (g^a, g^b, g^r)$
    - Goldwasser-Micali PKE ← QR assumption ← $y \leftarrow \mathbb{Z}_N^{\times}(+,+) \approx y \leftarrow \mathbb{Z}_N^{\times}(-,-)$

Recall from Last Lecture

- Task 4: Public-key encryption
  - Modelled setting and security (CPA secrecy)
  - Saw two CPA-secret constructions, with proofs:
    - ElGamal PKE ← DDH assumption $(g^a, g^b, g^{ab}) \approx (g^a, g^b, g^r)$
    - Goldwasser-Micali PKE ← QR assumption $y \leftarrow \mathbb{Z}_N^\times (+,+) \approx y \leftarrow \mathbb{Z}_N^\times (-,-)$
  - Conceptual takeaways:
    1. Two-message key-exchange ↔ PKE
    2. Structure vs. hardness

Two ways to generate the same "OTP" $(g^a)^b \rightarrow g^{ab} \leftarrow (g^b)^a$

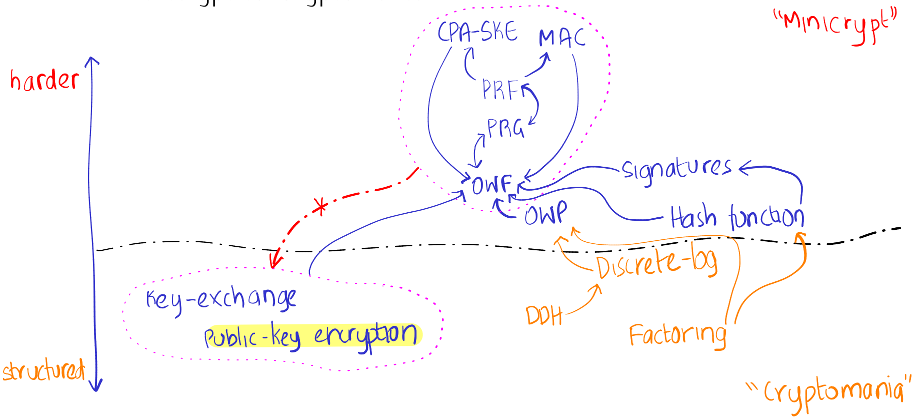
Recall from Last Lecture

- Task 4: Public-key encryption
  - Modelled setting and security (CPA secrecy)
  - Saw two CPA-secret constructions, with proofs:
    - ElGamal PKE ← DDH assumption ← $(g^a, g^b, g^{ab}) \approx (g^a, g^b, g^r)$
    - Goldwasser-Micali PKE ← QR assumption

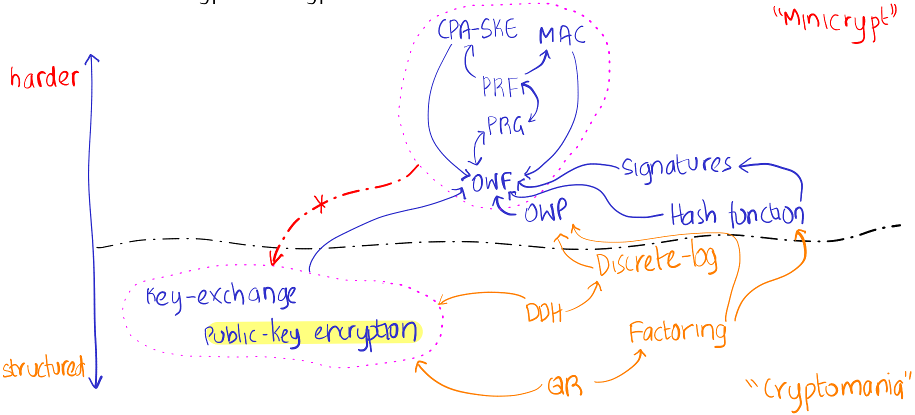
      ↖ $y \leftarrow \mathbb{Z}_N^\times (+,+) \approx y \leftarrow \mathbb{Z}_N^\times (-,-)$
  - Conceptual takeaways:
    1. Two-message key-exchange ↔ PKE
    2. Structure vs. hardness

       ↖ Two ways to generate the same "OTP" $(g^a)^b \rightarrow g^{ab} \leftarrow (g^b)^a$
- Some open questions:
  1. CPA-PKE $\overset{?}{\rightarrow}$ CCA-PKE
     - Recall that CPA-SKE → CCA-SKE!
  2. DLog $\overset{?}{\rightarrow}$ CPA-PKE
     - We know CDH → CPA-PKE in the "random-oracle model"

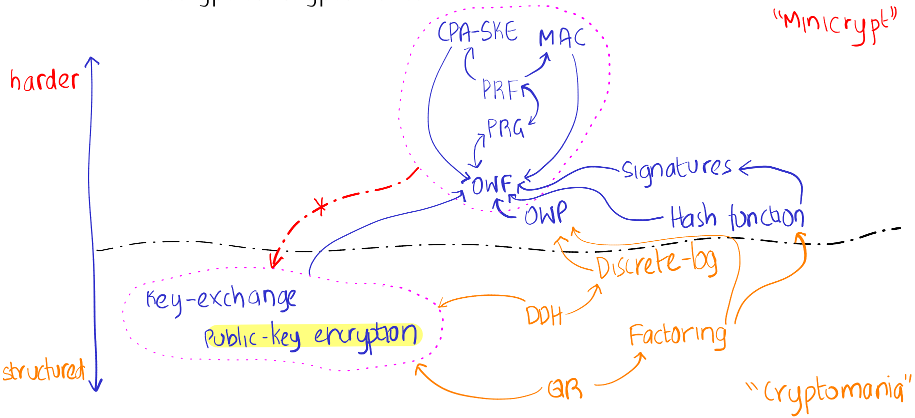
- Minicrypt to Cryptomania



- Minicrypt to Cryptomania



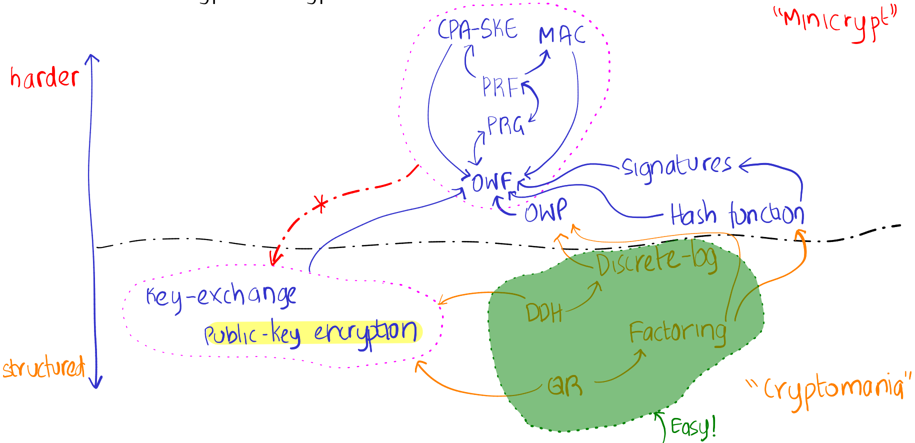
- Minicrypt to Cryptomania



- Today: Task 4 against stronger class of *quantum* Eves

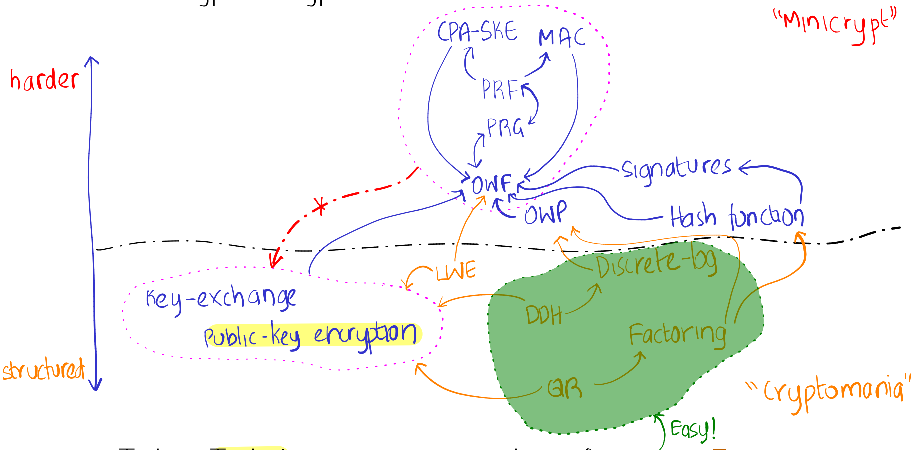
Plan for This Lecture...

- Minicrypt to Cryptomania



- Today: Task 4 against stronger class of *quantum* Eves

- Minicrypt to Cryptomania



- Today: Task 4 against stronger class of *quantum* Eves

Plan for This Lecture...

computational secrecy

General *template*:

1. Identify the task — Public-key encryption
2. Come up with precise threat model M (a.k.a security model) — Eavesdroppers
   - Adversary/Attack: What are the adversary's capabilities?
   - Security Goal: What does it mean to be secure?
3. Construct a scheme Π
4. Formally prove that Π in secure in model M

Plan for This Lecture...

quantum computational secrecy

General *template*:

Public-key encryption   Eavesdroppers

1. Identify the task
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities?
   - Security Goal: What does it mean to be secure?
3. Construct a scheme Π
4. Formally prove that Π in secure in model M

Plan for This Lecture...

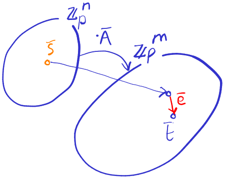
quantum computational secrecy

General *template*:

1. Identify the task → Public-key encryption
   Eavesdroppers
2. Come up with precise threat model M (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities?
   - Security Goal: What does it mean to be secure?
3. Construct a scheme Π ← Regev's PKE
4. Formally prove that Π in secure in model M
   Assuming quantum hardness of "learning with errors" (LWE)

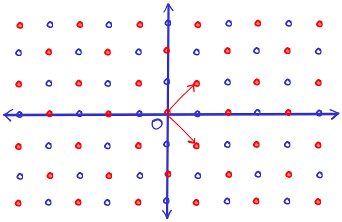
Plan for This Lecture...

1 Motivation: Quantum Adversaries



2 Learning with Errors (LWE)

3 Cryptography from LWE



4 LWE and Lattices

Plan for This Lecture...

1 Motivation: Quantum Adversaries

2 Learning with Errors (LWE)

3 Cryptography from LWE

4 LWE and Lattices

Quantum Computation

- Based on principles of quantum mechanics
  1. Certain physical properties (e.g., electron spin) are “discrete”
  2. Its value is a “superposition” of these discrete values

Quantum Computation

- Based on principles of quantum mechanics
  1. Certain physical properties (e.g., electron spin) are "discrete"
  2. Its value is a "superposition" of these discrete values

- *Classical* computation vs. Quantum computation

Quantum Computation

- Based on principles of quantum mechanics
  1. Certain physical properties (e.g., electron spin) are "discrete"
  2. Its value is a "superposition" of these discrete values

- *Classical* computation vs. Quantum computation
  1. Bits .... Qubits (Quantum bits)

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

$\mathbb{C}$

$|b\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle \; : \; \alpha_0^2 + \alpha_1^2 = 1$

Quantum Computation

- Based on principles of quantum mechanics
  1. Certain physical properties (e.g., electron spin) are "discrete"
  2. Its value is a "superposition" of these discrete values

- *Classical* computation vs. Quantum computation
  1. Bits .... Qubits (Quantum bits)

     $b \in \{0,1\}$ .... $|b\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle : \alpha_0^2 + \alpha_1^2 = 1$, $\alpha_0, \alpha_1 \in \mathbb{C}$

  2. Classical state .... Quantum state

     $\bar{b} = b_1 b_2 \ldots b_n \in \{0,1\}^n$ .... $\psi = \sum_{\bar{b} \in \{0,1\}^n} \alpha_{\bar{b}} |\bar{b}\rangle : \sum_{\bar{b} \in \{0,1\}^n} \alpha_{\bar{b}}^2 = 1$, $\alpha_{\bar{b}} \in \mathbb{C}$

Quantum Computation

- Based on principles of quantum mechanics
  1. Certain physical properties (e.g., electron spin) are "discrete"
  2. Its value is a "superposition" of these discrete values

- *Classical* computation vs. Quantum computation
  1. Bits .... Qubits (Quantum bits)

     $b \in \{0,1\}$ .... $|b\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle : \alpha_0^2 + \alpha_1^2 = 1$ ($\alpha_0, \alpha_1 \in \mathbb{C}$)
  2. Classical state .... Quantum state

     $\bar{b} = b_1 b_2 \ldots b_n \in \{0,1\}^n$ .... $\psi = \sum_{\bar{b} \in \{0,1\}^n} \alpha_{\bar{b}} |\bar{b}\rangle : \sum_{\bar{b} \in \{0,1\}^n} \alpha_{\bar{b}}^2 = 1$ ($\alpha_{\bar{b}} \in \mathbb{C}$)
  3. Classical circuit .... Quantum circuits

     $\bar{b} \rightarrow \boxed{C : \{0,1\}^n \rightarrow \{0,1\}^n} \rightarrow \bar{b}'$ .... $\psi \rightarrow \boxed{U : \{\alpha_{\bar{b}}\}_{\bar{b}} \rightarrow \{\alpha'_{\bar{b}}\}_{\bar{b}}} \rightarrow \psi'$

Quantum Computation

- Based on principles of quantum mechanics
  1. Certain physical properties (e.g., electron spin) are “discrete”
  2. Its value is a “superposition” of these discrete values

- *Classical* computation vs. Quantum computation
  1. Bits .... Qubits (Quantum bits)

     $b \in \{0,1\}$ $\qquad$ $|b\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle \; : \; \alpha_0^2 + \alpha_1^2 = 1$ $\qquad$ $\mathbb{C}$

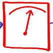
  2. Classical state .... Quantum state

     $\bar{b} = b_1 b_2 \ldots b_n \in \{0,1\}^n$ $\qquad$ $\psi = \sum_{\bar{b}\in\{0,1\}^n} \alpha_{\bar{b}} |\bar{b}\rangle \; : \; \sum_{\bar{b}\in\{0,1\}^n} \alpha_{\bar{b}}^2 = 1$ $\qquad$ $\mathbb{C}$

  3. Classical circuit $\qquad$ Quantum circuits

     $\bar{b} \to \boxed{C : \{0,1\}^n \to \{0,1\}^n} \to \bar{b}'$ $\qquad$ $\psi \to \boxed{U : \{\alpha_{\bar{b}}\}_{\bar{b}} \to \{\alpha'_{\bar{b}}\}_{\bar{b}}} \to \psi'$

  4. .... Measurement $\leftarrow$ randomised

     $\psi \to$  $\to \bar{b}$

Quantum Computation

- Based on principles of quantum mechanics
  1. Certain physical properties (e.g., electron spin) are “discrete”
  2. Its value is a “superposition” of these discrete values

- *Classical* computation vs. Quantum computation
  1. Bits .... Qubits (Quantum bits)

     $b \in \{0,1\}$ $\qquad$ $\mathbb{C}$ $\qquad$ $|b\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle : \alpha_0^2 + \alpha_1^2 = 1$
  2. Classical state .... Quantum state

     $\bar{b} = b_1 b_2 \dots b_n \in \{0,1\}^n$ $\qquad$ $\mathbb{C}$ $\qquad$ $\psi = \sum_{\bar{b} \in \{0,1\}^n} \alpha_{\bar{b}} |\bar{b}\rangle : \sum_{\bar{b} \in \{0,1\}^n} \alpha_{\bar{b}}^2 = 1$
  3. Classical circuit $\qquad$ Quantum circuits

     $\bar{b} \rightarrow \boxed{C : \{0,1\}^n \rightarrow \{0,1\}^n} \rightarrow \bar{b}'$ $\qquad$ $\psi \rightarrow \boxed{U : \{\alpha_{\bar{b}}\}_{\bar{b}} \rightarrow \{\alpha'_{\bar{b}}\}_{\bar{b}}} \rightarrow \psi'$
  4. .... Measurement ← randomised

     $\psi \rightarrow \boxed{\ } \rightarrow \bar{b}$

     
  5. PPT adversary .... Quantum PT adversary

Modelling the Setting for Quantum Adversaries...

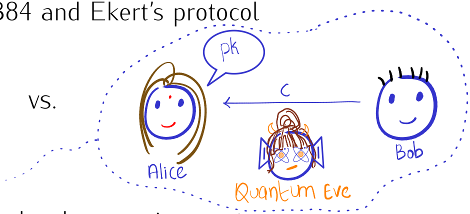
- Cryptography in a quantum world (quantum cryptography)
  - All parties have access to quantum computers and channel

Modelling the Setting for Quantum Adversaries...

- Cryptography in a quantum world (quantum cryptography)
  - All parties have access to quantum computers and channel
  - E.g.: key-exchange possible assuming only *authenticated* classical channel: see BB84 and Ekert's protocol

- Cryptography in a quantum world (quantum cryptography)
  - All parties have access to quantum computers and channel
  - E.g.: key-exchange possible assuming only *authenticated* classical channel: see BB84 and Ekert's protocol

vs.

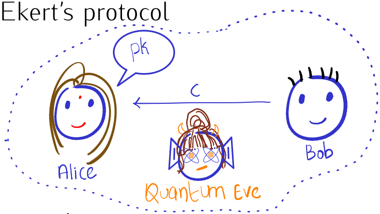


- *Post-quantum* cryptography
  - Honest parties are classical; adversary is quantum

Modelling the Setting for Quantum Adversaries...

- Cryptography in a quantum world (quantum cryptography)
  - All parties have access to quantum computers and channel
  - E.g.: key-exchange possible assuming only *authenticated* classical channel: see BB84 and Ekert's protocol

vs.



- *Post-quantum* cryptography
  - Honest parties are classical; adversary is quantum
  - Possible attack scenario: "Harvest now, decrypt later"
    - Potential adversaries: Five Eyes, state actors...

Modelling the Setting for Quantum Adversaries...

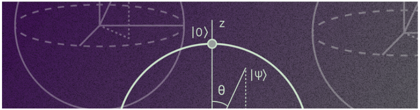
NEWS

NIST Releases First 3 Finalized Post-Quantum Encryption Standards

August 13, 2024

Quantum Resistance and the Signal Protocol

ehrenkret on 19 Sep 2023



 Security Research

February 21, 2024

iMessage with PQ3: The new state of the art in quantum-secure messaging at scale

Posted by Apple Security Engineering and Architecture (SEAR)

विज्ञान एवं प्रौद्योगिकी विभाग
DEPARTMENT OF
SCIENCE & TECHNOLOGY

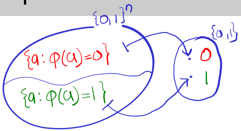
सत्यमेव जयते

National Quantum Mission (NQM)

The Union Cabinet, approved the National Quantum Mission (NQM) on 19th April 2023 at a total cost of Rs.6003.65 crore from 2023-24 to 2030-31, aiming to seed, nurture and scale up scientific and industrial R&D and create a vibrant & innovative ecosystem in Quantum Technology (QT). This will accelerate QT

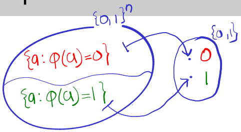
- Recent effort to research/deploy post-quantum cryptography

What is Easier for Quantum Computers?...

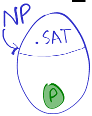


- *Unstructured* search problem:
  - Input: n-variable Boolean formula φ
  - Solution: a *satisfying* assignment $a \in \{0,1\}^n : \varphi(a) = 1$

What is Easier for Quantum Computers?

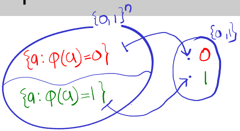


- *Unstructured* search problem:
  - Input: n-variable Boolean formula φ
  - Solution: a *satisfying* assignment $a \in \{0,1\}^n : \varphi(a) = 1$

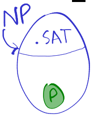


- Classical setting:
  - NP complete (SAT)
  - *Sub-exponential-time* algorithms believed to not exist (exponential-time hypothesis)

What is Easier for Quantum Computers?



- *Unstructured* search problem:
  - Input: n-variable Boolean formula φ
  - Solution: a *satisfying* assignment $a \in \{0,1\}^n : \varphi(a) = 1$
- Classical setting:

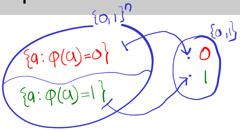


  - NP complete (SAT)
  - *Sub-exponential-time* algorithms believed to not exist (exponential-time hypothesis)
- Quantum setting:

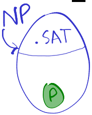
Theorem 1 (Grover's algorithm)

There is a quantum algorithm that given φ (represented as a classical circuit) finds a satisfying assignment in time $2^{O(n/2)}$

What is Easier for Quantum Computers?



- *Unstructured* search problem:
  - Input: n-variable Boolean formula φ
  - Solution: a *satisfying* assignment $a \in \{0,1\}^n : \varphi(a) = 1$
- Classical setting:
  - NP complete (SAT)
  - *Sub-exponential-time* algorithms believed to not exist (exponential-time hypothesis)



- Quantum setting:

Theorem 1 (Grover's algorithm)

There is a quantum algorithm that given φ (represented as a classical circuit) finds a satisfying assignment in time $2^{O(n/2)}$

- Impact on cryptography: SKEs broken in *quantum* time $2^{O(n/2)}$
  - Solution: double key-size (use 256-bit AES instead of 128-bit)

What is Easier for Quantum Computers?...

Addition modulo prime p

$0 = p \to 1 \to 2 \to \cdots \to i \to \cdots \to p-1 \to 0$

$(\mathbb{Z}_p, +)$

$\{0, \dots, p-1\}$ $\quad$ $g_1 + g_2 := g_1 + g_2 \pmod p$

- order p
- cyclic

Multiplication modulo prime p

$1 = g^{p-1} \bmod p \to g^1 \bmod p \to \cdots \to g^i \bmod p \to \cdots \to 1$

$(\mathbb{Z}_p^\times, \cdot)$

$\{1, \dots, p-1\}$ $\quad$ $g_1 \cdot g_2 := g_1 \cdot g_2 \pmod p$

- order $p-1$
- cyclic

Multiplication modulo $N = pq$

primes

$(\mathbb{Z}_N^\times, \cdot)$

$\{1, \dots, p-1,$
$p+1, \dots, 2p-1,$
$\dots, pq-1\}$

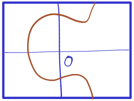
$g_1 \cdot g_2 := g_1 \cdot g_2 \pmod N$

- order $(p-1)(q-1)$
- not cyclic

Elliptic curves modulo prime p

$(E, +)$ "curve points addition"

solutions to
$y^2 = x^3 + Ax + B \pmod p$



- $|p+1 - \text{order}| \leq 2\sqrt{p}$
- cyclic

What is Easier for Quantum Computers?...

DLog easy

Addition modulo prime p

$0 = p \to 1 \to 2 \to \cdots \to i \to \cdots \to p-1 \to 0$

$(\mathbb{Z}_p, +)$

$\{0, \dots p-1\}$ $\quad g_1 + g_2 := g_1 + g_2 \pmod p$

- order p
- cyclic

DLog hard, but DDH easy

Multiplication modulo prime p

$1 = g^{p-1} \bmod p \to g^1 \bmod p \to \cdots \to g^i \bmod p \to \cdots \to 1$

$(\mathbb{Z}_p^\times, \cdot)$

$\{1, \dots p-1\}$ $\quad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod p$

- order $p-1$
- cyclic

Multiplication modulo $N = pq$

primes

$(\mathbb{Z}_N^\times, \cdot)$

$\{1, \dots p-1, p+1, \dots, 2p-1, \dots pq-1\}$ $\quad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod N$

- order $(p-1)(q-1)$
- not cyclic

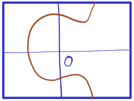
DDH hard in "subgroup"

Elliptic curves modulo prime p

$(E, +)$ "curve points addition"

solutions to

$y^2 = x^3 + Ax + B \pmod p$

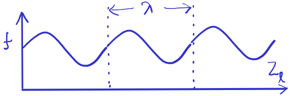


- $|p+1 - \text{order}| \leq 2\sqrt{p}$
- cyclic

DLog very hard, DDH hard

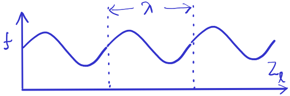
What is Easier for Quantum Computers?...

- Structured *period-finding* problem for functions over $(\mathbb{Z}_\ell, +)$
  - Input: $f : (\mathbb{Z}_\ell, +) \to \mathbb{G}$ that is "periodic"
    - That is, $\exists \lambda \in \mathbb{Z}_\ell \forall x \in \mathbb{Z}_\ell : f(x + \lambda) = f(x)$



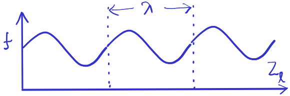
What is Easier for Quantum Computers?...

- Structured *period-finding* problem for functions over $(\mathbb{Z}_\ell, +)$
  - Input: $\mathrm{f} : (\mathbb{Z}_\ell, +) \to \mathbb{G}$ that is “periodic”
    - That is, $\exists \lambda \in \mathbb{Z}_\ell \forall x \in \mathbb{Z}_\ell : \mathrm{f}(x + \lambda) = \mathrm{f}(x)$
  - Solution: smallest “period” λ



What is Easier for Quantum Computers?...

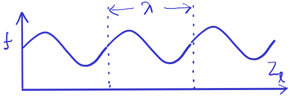
- Structured *period-finding* problem for functions over $(\mathbb{Z}_\ell, +)$
  - Input: $f : (\mathbb{Z}_\ell, +) \to \mathbb{G}$ that is "periodic"
    - That is, $\exists \lambda \in \mathbb{Z}_\ell \forall x \in \mathbb{Z}_\ell : f(x + \lambda) = f(x)$
  - Solution: smallest "period" λ



- Classical setting: *PPT* algorithms believed not to exist for certain fs.

What is Easier for Quantum Computers?...

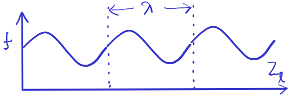
- Structured *period-finding* problem for functions over $(\mathbb{Z}_\ell, +)$
  - Input: $f : (\mathbb{Z}_\ell, +) \to \mathbb{G}$ that is "periodic"
    - That is, $\exists \lambda \in \mathbb{Z}_\ell \forall x \in \mathbb{Z}_\ell : f(x + \lambda) = f(x)$
  - Solution: smallest "period" λ



- Classical setting: *PPT* algorithms believed not to exist for certain fs. E.g.:
  1. $f_{a,N}(x) := a^x \bmod N$, where $\mathbb{G} = (\mathbb{Z}_N^\times, \cdot)$ and $a \leftarrow \mathbb{Z}_N^\times$
     - ? What is the period of $f_{a,N}$?

What is Easier for Quantum Computers?...

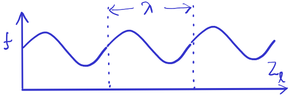
- Structured *period-finding* problem for functions over $(\mathbb{Z}_\ell, +)$
  - Input: $\mathrm{f} : (\mathbb{Z}_\ell, +) \to \mathbb{G}$ that is "periodic"
    - That is, $\exists \lambda \in \mathbb{Z}_\ell \forall x \in \mathbb{Z}_\ell : \mathrm{f}(x + \lambda) = \mathrm{f}(x)$
  - Solution: smallest "period" λ



- Classical setting: *PPT* algorithms believed not to exist for certain fs. E.g.:
  1. $\mathrm{f}_{a,N}(x) := a^x \bmod N$, where $\mathbb{G} = (\mathbb{Z}_N^\times, \cdot)$ and $a \leftarrow \mathbb{Z}_N^\times$
     - ? What is the period of $\mathrm{f}_{a,N}$? $\lambda(N) := (p-1)(q-1)/2$ (w.h.p.)
     - Finding $\lambda(N)$ equivalent to factoring N

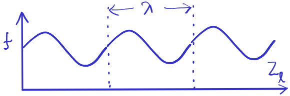
What is Easier for Quantum Computers?...

- Structured *period-finding* problem for functions over $(\mathbb{Z}_\ell, +)$
  - Input: $f : (\mathbb{Z}_\ell, +) \to \mathbb{G}$ that is "periodic"
    - That is, $\exists \lambda \in \mathbb{Z}_\ell \forall x \in \mathbb{Z}_\ell : f(x + \lambda) = f(x)$
  - Solution: smallest "period" λ



- Classical setting: *PPT* algorithms believed not to exist for certain fs. E.g.:
  1. $f_{a,N}(x) := a^x \bmod N$, where $\mathbb{G} = (\mathbb{Z}_N^\times, \cdot)$ and $a \leftarrow \mathbb{Z}_N^\times$
     - ? What is the period of $f_{a,N}$? $\lambda(N) := (p - 1)(q - 1)/2$ (w.h.p.)
     - Finding $\lambda(N)$ equivalent to factoring N
  2. $f_{g,h}(x, y) := g^x h^{-y} \bmod p$, where $\mathbb{G} = (\mathbb{Z}_p^\times, \cdot)$ and $g, h \leftarrow \mathbb{Z}_p^\times$
     - ? What is the period of $f_{g,h}$?

- Structured *period-finding* problem for functions over $(\mathbb{Z}_\ell, +)$
  - Input: $\mathrm{f} : (\mathbb{Z}_\ell, +) \to \mathbb{G}$ that is "periodic"
    - That is, $\exists \lambda \in \mathbb{Z}_\ell \forall x \in \mathbb{Z}_\ell : \mathrm{f}(x + \lambda) = \mathrm{f}(x)$
  - Solution: smallest "period" λ

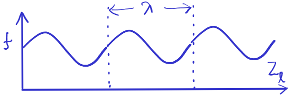


- Classical setting: *PPT* algorithms believed not to exist for certain fs. E.g.:
  1. $\mathrm{f}_{a,N}(x) := a^x \bmod N$, where $\mathbb{G} = (\mathbb{Z}_N^\times, \cdot)$ and $a \leftarrow \mathbb{Z}_N^\times$
     - ? What is the period of $\mathrm{f}_{a,N}$? $\lambda(N) := (p-1)(q-1)/2$ (w.h.p.)
     - Finding $\lambda(N)$ equivalent to factoring N
  2. $\mathrm{f}_{g,h}(x, y) := g^x h^{-y} \bmod p$, where $\mathbb{G} = (\mathbb{Z}_p^\times, \cdot)$ and $g, h \leftarrow \mathbb{Z}_p^\times$
     - ? What is the period of $\mathrm{f}_{g,h}$? $\lambda(g, h) := (\log_g(h), 1)$, the discrete log!

$$f_{g,h}(x+\log_g h, y+1) = g^{x+\log_g h} \cdot h^{-y-1} = g^x \cdot \cancel{h} \cdot h^{-y} \cdot \cancel{h^{-1}} = g^x h^{-y} = f_{g,h}(x,y)$$

What is Easier for Quantum Computers?...

- Structured *period-finding* problem for functions over $(\mathbb{Z}_\ell, +)$
  - Input: $f : (\mathbb{Z}_\ell, +) \to \mathbb{G}$ that is "periodic"
    - That is, $\exists \lambda \in \mathbb{Z}_\ell \forall x \in \mathbb{Z}_\ell : f(x+\lambda) = f(x)$
  - Solution: smallest "period" λ

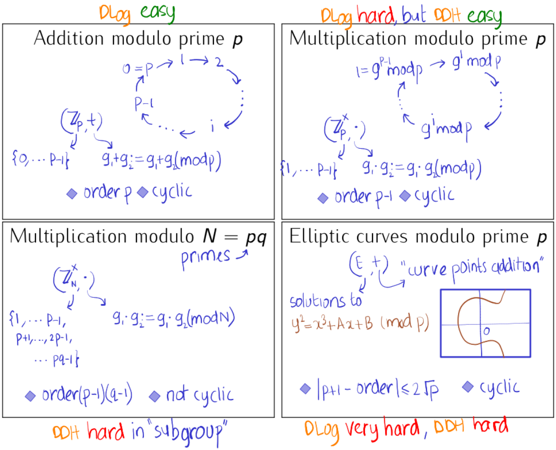


- Classical setting: *PPT* algorithms believed not to exist for certain fs. E.g.:
  1. $f_{a,N}(x) := a^x \bmod N$, where $\mathbb{G} = (\mathbb{Z}_N^\times, \cdot)$ and $a \leftarrow \mathbb{Z}_N^\times$
     - ? What is the period of $f_{a,N}$? $\lambda(N) := (p-1)(q-1)/2$ (w.h.p.)
     - Finding $\lambda(N)$ equivalent to factoring N
  2. $f_{g,h}(x,y) := g^x h^{-y} \bmod p$, where $\mathbb{G} = (\mathbb{Z}_p^\times, \cdot)$ and $g, h \leftarrow \mathbb{Z}_p^\times$
     - ? What is the period of $f_{g,h}$? $\lambda(g,h) := (\log_g(h), 1)$, the discrete log!
- Quantum setting: $f_{g,h}(x+\log_g h, y+1) = g^{x+\log_g h} \cdot h^{-y-1} = g^x \cdot \cancel{h} \cdot h^{-y} \cdot \cancel{h^{-1}} = g^x h^{-y} = f_{g,h}(x,y)$

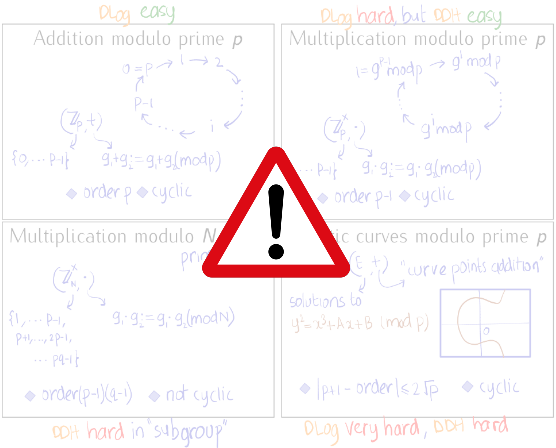
Theorem 2 (Shor's algorithm)

There is a quantum algorithm that finds the period λ of a periodic function f as above (represented as a classical circuit) in time polynomial in $|\mathbb{Z}_\ell| = \log(\ell)$.

What is Easier for Quantum Computers?...

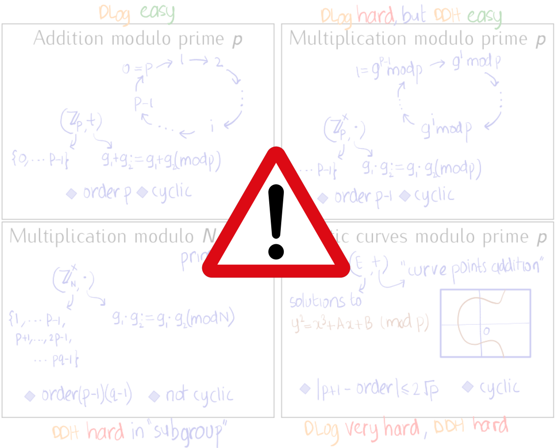


- Corollary: factoring and discrete log are quantum *easy*!



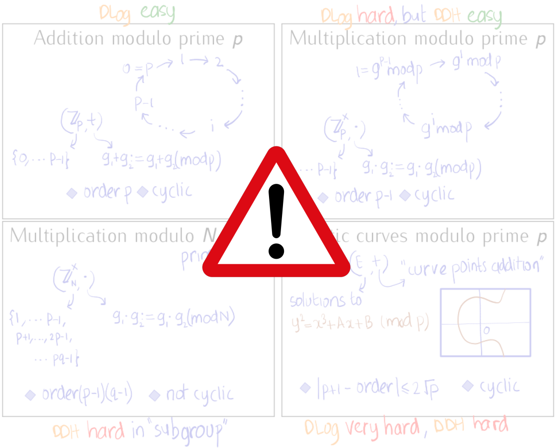
- Corollary: factoring and discrete log are quantum *easy*!

What is Easier for Quantum Computers?...



- Corollary: factoring and discrete log are quantum *easy*!
- Impact on cryptography: PKEs from previous lecture insecure!

What is Easier for Quantum Computers?...

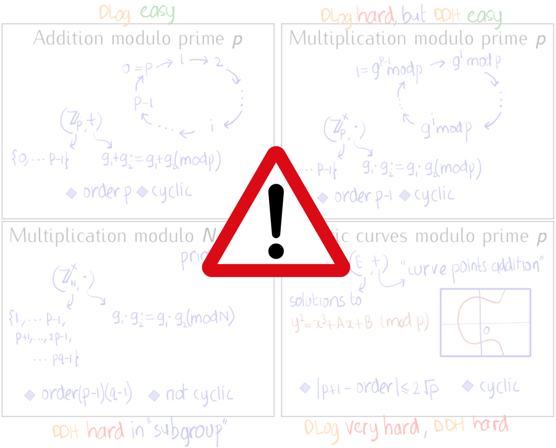


- Corollary: factoring and discrete log are quantum *easy*!

Impact on cryptography: PKEs from previous lecture insecure!

  - We need: hardness assumption that holds against QPT...

What is Easier for Quantum Computers?...



- Corollary: factoring and discrete log are quantum *easy*!

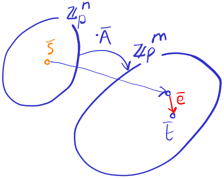
Impact on cryptography: PKEs from previous lecture insecure!

  - We need: hardness assumption that holds against QPT...
  - ...that has sufficient structure to allow PKE/key exchange

Plan for this Lecture

1 Motivation: Quantum Adversaries

2 Learning with Errors (LWE)



3 Cryptography from LWE

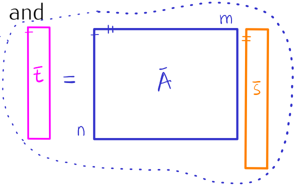
4 LWE and Lattices

Solving Linear Equations Over $(\mathbb{Z}_p, +, \cdot)$...

- Let's consider $(\mathbb{Z}_p, +, \cdot)$, i.e., $(\mathbb{Z}_p, +)$ with multiplication over $\mathbb{Z}_p^\times$

Solving Linear Equations Over $(\mathbb{Z}_p, +, \cdot)$...

- Let's consider $(\mathbb{Z}_p, +, \cdot)$, i.e., $(\mathbb{Z}_p, +)$ with multiplication over $\mathbb{Z}_p^\times$
- Candidates:
  1. Solve system of random linear equations over $(\mathbb{Z}_p, +, \cdot)$?
     - Input: $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}, \bar{s} \leftarrow \mathbb{Z}_p^m$ and $\bar{t} := \bar{A}\bar{s} \bmod p$



- Let's consider $(\mathbb{Z}_p, +, \cdot)$, i.e., $(\mathbb{Z}_p, +)$ with multiplication over $\mathbb{Z}_p^\times$
- Candidates:
  1. Solve system of random linear equations over $(\mathbb{Z}_p, +, \cdot)$?
     - Input: $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}$, $\bar{s} \leftarrow \mathbb{Z}_p^m$ and

$$\bar{t} := \bar{A}\bar{s} \bmod p$$

$$\begin{bmatrix} t_1 \\ \vdots \\ t_n \end{bmatrix} = \begin{array}{cc} & m \\ n & \begin{bmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{mn} \end{bmatrix} \end{array} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{bmatrix}$$

Solving Linear Equations Over $(\mathbb{Z}_p, +, \cdot)$...

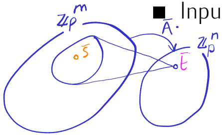
- Let's consider $(\mathbb{Z}_p, +, \cdot)$, i.e., $(\mathbb{Z}_p, +)$ with multiplication over $\mathbb{Z}_p^\times$
- Candidates:
  1. Solve system of random linear equations over $(\mathbb{Z}_p, +, \cdot)$? poly(n)
     - Input: $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}, \bar{s} \leftarrow \mathbb{Z}_p^m$ and

$$\bar{t} := \bar{A}\bar{s} \bmod p$$

$$\begin{bmatrix} t_1 \\ \vdots \\ t_n \end{bmatrix} = \begin{array}{c} \;\; m \\ n \begin{bmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{mn} \end{bmatrix} \end{array} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{bmatrix}$$

Solving Linear Equations Over $(\mathbb{Z}_p, +, \cdot)$...

- Let's consider $(\mathbb{Z}_p, +, \cdot)$, i.e., $(\mathbb{Z}_p, +)$ with multiplication over $\mathbb{Z}_p^\times$
- Candidates:
  1. Solve system of random linear equations over $(\mathbb{Z}_p, +, \cdot)$? poly(n)
     - Input: $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}$, $\bar{s} \leftarrow \mathbb{Z}_p^m$ and

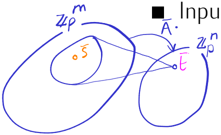


$$\bar{t} := \bar{A}\bar{s} \bmod p$$

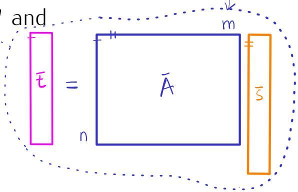
$$\begin{bmatrix} t_1 \\ \vdots \\ t_n \end{bmatrix} = \begin{array}{c} m \\ n \begin{bmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{mn} \end{bmatrix} \end{array} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{bmatrix}$$

Solving Linear Equations Over $(\mathbb{Z}_p, +, \cdot)$...

- Let's consider $(\mathbb{Z}_p, +, \cdot)$, i.e., $(\mathbb{Z}_p, +)$ with multiplication over $\mathbb{Z}_p^\times$
- Candidates:
  1. Solve system of random linear equations over $(\mathbb{Z}_p, +, \cdot)$? poly(n)
     - Input: $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}, \bar{s} \leftarrow \mathbb{Z}_p^m$ and



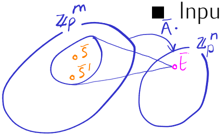
$$\bar{t} := \bar{A}\bar{s} \bmod p$$



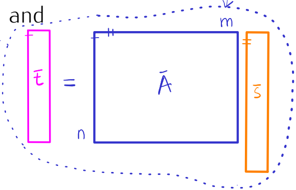
     - Solution: $\bar{s}$?

Solving Linear Equations Over $(\mathbb{Z}_p, +, \cdot)$...

- Let's consider $(\mathbb{Z}_p, +, \cdot)$, i.e., $(\mathbb{Z}_p, +)$ with multiplication over $\mathbb{Z}_p^\times$
- Candidates:
  1. Solve system of random linear equations over $(\mathbb{Z}_p, +, \cdot)$? poly(n)
     - Input: $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}, \bar{s} \leftarrow \mathbb{Z}_p^m$ and



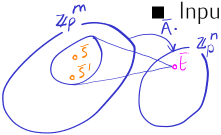
$$\bar{t} := \bar{A}\bar{s} \bmod p$$



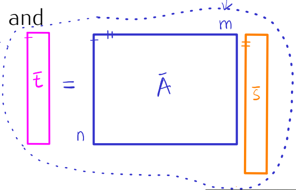
     - Solution: $\bar{s}$?
     - Problem: Information-theoretically hard!
     - Solution: *some* preimage $\bar{s}'$ of $\bar{t}$?

Solving Linear Equations Over $(\mathbb{Z}_p, +, \cdot)$...

- Let's consider $(\mathbb{Z}_p, +, \cdot)$, i.e., $(\mathbb{Z}_p, +)$ with multiplication over $\mathbb{Z}_p^\times$
- Candidates:
  1. Solve system of random linear equations over $(\mathbb{Z}_p, +, \cdot)$? poly(n)
     - Input: $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}$, $\bar{s} \leftarrow \mathbb{Z}_p^m$ and



$$\bar{t} := \bar{A}\bar{s} \bmod p$$

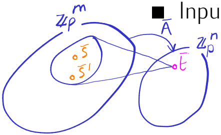


     - Solution: $\bar{s}$?
     - Problem: Information-theoretically hard!
     - Solution: *some* preimage $\bar{s}'$ of $\bar{t}$?
     - Problem: Solvable in polynomial time: Gaussian elimination

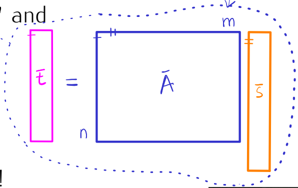
Solving Linear Equations Over $(\mathbb{Z}_p, +, \cdot)$...

- Let's consider $(\mathbb{Z}_p, +, \cdot)$, i.e., $(\mathbb{Z}_p, +)$ with multiplication over $\mathbb{Z}_p^{\times}$
- Candidates:

1. Solve system of random linear equations over $(\mathbb{Z}_p, +, \cdot)$? poly(n)
   - Input: $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}$, $\bar{s} \leftarrow \mathbb{Z}_p^{m}$ and



$$\bar{t} := \bar{A}\bar{s} \bmod p$$



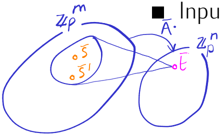
   - Solution: $\bar{s}$?
   - Problem: Information-theoretically hard!
   - Solution: *some* preimage $\bar{s}'$ of $\bar{t}$?
   - Problem: Solvable in polynomial time: Gaussian elimination

2. Solve system of random linear equations over $(\mathbb{Z}_p, +, \cdot)$?
   - Input: $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}$, $\bar{s} \leftarrow \mathbb{Z}_p^{n}$ and

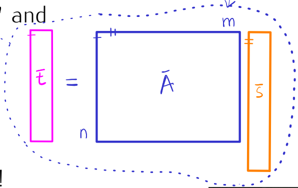
$$\bar{t}^{\top} := \bar{s}^{\top}\bar{A} \bmod p$$

   - Solution: $\bar{s}$?

- Let's consider $(\mathbb{Z}_p, +, \cdot)$, i.e., $(\mathbb{Z}_p, +)$ with multiplication over $\mathbb{Z}_p^\times$
- Candidates:
  1. Solve system of random linear equations over $(\mathbb{Z}_p, +, \cdot)$? poly(n)
     - Input: $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}, \bar{s} \leftarrow \mathbb{Z}_p^m$ and



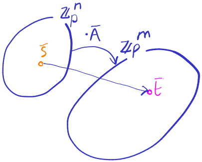
$$\bar{t} := \bar{A}\bar{s} \bmod p$$



     - Solution: $\bar{s}$?
     - Problem: Information-theoretically hard!
     - Solution: *some* preimage $\bar{s}'$ of $\bar{t}$?
     - Problem: Solvable in polynomial time: Gaussian elimination

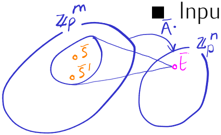
  2. Solve system of random linear equations over $(\mathbb{Z}_p, +, \cdot)$?
     - Input: $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}, \bar{s} \leftarrow \mathbb{Z}_p^n$ and

$$\bar{t}^\top := \bar{s}^\top \bar{A} \bmod p$$

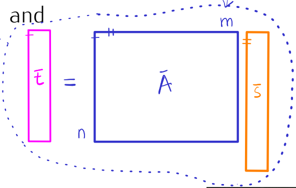


     - Solution: $\bar{s}$?

- Let's consider $(\mathbb{Z}_p, +, \cdot)$, i.e., $(\mathbb{Z}_p, +)$ with multiplication over $\mathbb{Z}_p^\times$
- Candidates:
  1. Solve system of random linear equations over $(\mathbb{Z}_p, +, \cdot)$? poly(n)
     - Input: $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}, \bar{s} \leftarrow \mathbb{Z}_p^m$ and



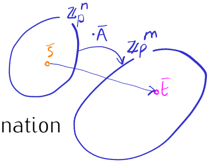
$$\bar{t} := \bar{A}\bar{s} \bmod p$$



     - Solution: $\bar{s}$?
     - Problem: Information-theoretically hard!
     - Solution: *some* preimage $\bar{s}'$ of $\bar{t}$?
     - Problem: Solvable in polynomial time: Gaussian elimination

  2. Solve system of random linear equations over $(\mathbb{Z}_p, +, \cdot)$?
     - Input: $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}, \bar{s} \leftarrow \mathbb{Z}_p^n$ and

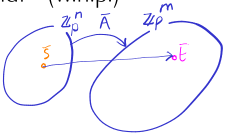
$$\bar{t}^\top := \bar{s}^\top \bar{A} \bmod p$$



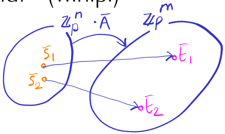
     - Solution: $\bar{s}$?
     - Problem: Still solvable by Gaussian elimination

Solving Linear Equations Over $(\mathbb{Z}_p, +, \cdot)$...

- The map $\bar{s} \mapsto (\bar{s}^\top \bar{A})^\top$ is a "random linear code"
  - Two "codewords" $\bar{t}_1^\top := \bar{s}_1^\top \bar{A}$ and $\bar{t}_2^\top := \bar{s}_2^\top \bar{A}$ are "far" (w.h.p.)
  - However, efficient "decoding" algorithm to recover $\bar{s}$ from "noisy" $\bar{t}^\top \approx \bar{s}^\top \bar{A}$ *not known*

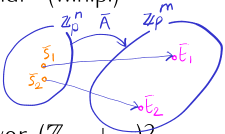


- The map $\bar{s} \mapsto (\bar{s}^\top \bar{A})^\top$ is a "random linear code"
  - Two "codewords" $\bar{t}_1^\top := \bar{s}_1^\top \bar{A}$ and $\bar{t}_2^\top := \bar{s}_2^\top \bar{A}$ are "far" (w.h.p.)
  - However, efficient "decoding" algorithm to recover $\bar{s}$ from "noisy" $\bar{t}^\top \approx \bar{s}^\top \bar{A}$ *not known*



Solving Linear Equations Over $(\mathbb{Z}_p, +, \cdot)$...

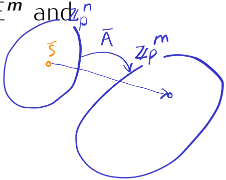
- The map $\bar{s} \mapsto (\bar{s}^\top \bar{A})^\top$ is a "random linear code"
  - Two "codewords" $\bar{t}_1^\top := \bar{s}_1^\top \bar{A}$ and $\bar{t}_2^\top := \bar{s}_2^\top \bar{A}$ are "far" (w.h.p.)
  - However, efficient "decoding" algorithm to recover $\bar{s}$ from "noisy" $\bar{t}^\top \approx \bar{s}^\top \bar{A}$ *not known*



3 Potentially hard: solve "noisy" linear equations over $(\mathbb{Z}_p, +, \cdot)$?

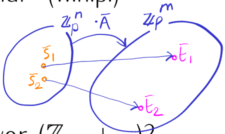
- Input $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}$, $\bar{s} \leftarrow \mathbb{Z}_p^n$, $\bar{e} \leftarrow \mathrm{E}^m$ and

$$\bar{t}^\top := \bar{s}^\top \bar{A} + \bar{e}^\top \bmod p$$



- Solution: $\bar{s}$

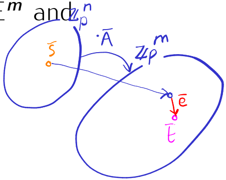
- The map $\bar{s} \mapsto (\bar{s}^\top \bar{A})^\top$ is a "random linear code"
  - Two "codewords" $\bar{t}_1^\top := \bar{s}_1^\top \bar{A}$ and $\bar{t}_2^\top := \bar{s}_2^\top \bar{A}$ are "far" (w.h.p.)
  - However, efficient "decoding" algorithm to recover $\bar{s}$ from "noisy" $\bar{t}^\top \approx \bar{s}^\top \bar{A}$ *not known*



3 Potentially hard: solve "noisy" linear equations over $(\mathbb{Z}_p, +, \cdot)$?

- Input $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}$, $\bar{s} \leftarrow \mathbb{Z}_p^n$, $\bar{e} \leftarrow \mathrm{E}^m$ and

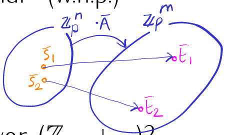
$$\bar{t}^\top := \bar{s}^\top \bar{A} + \bar{e}^\top \bmod p$$



- Solution: $\bar{s}$

Solving Linear Equations Over $(\mathbb{Z}_p, +, \cdot)$...

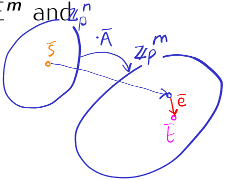
- The map $\bar{s} \mapsto (\bar{s}^\top \bar{A})^\top$ is a "random linear code"
  - Two "codewords" $\bar{t}_1^\top := \bar{s}_1^\top \bar{A}$ and $\bar{t}_2^\top := \bar{s}_2^\top \bar{A}$ are "far" (w.h.p.)
  - However, efficient "decoding" algorithm to recover $\bar{s}$ from "noisy" $\bar{t}^\top \approx \bar{s}^\top \bar{A}$ *not known*

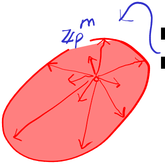


3. Potentially hard: solve "noisy" linear equations over $(\mathbb{Z}_p, +, \cdot)$?

   - Input $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}, \bar{s} \leftarrow \mathbb{Z}_p^n, \bar{e} \leftarrow \mathrm{E}^m$ and

$$\bar{t}^\top := \bar{s}^\top \bar{A} + \bar{e}^\top \bmod p$$

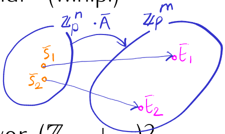




   - Solution: $\bar{s}$
   - Uninteresting case: E = uniform over $\mathbb{Z}_p$
     - $\bar{t}$ loses information about $\bar{s}$

Solving Linear Equations Over $(\mathbb{Z}_p, +, \cdot)$...

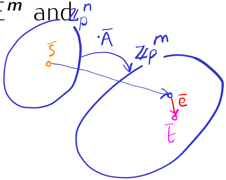
- The map $\bar{s} \mapsto (\bar{s}^\top \bar{A})^\top$ is a "random linear code"
  - Two "codewords" $\bar{t}_1^\top := \bar{s}_1^\top \bar{A}$ and $\bar{t}_2^\top := \bar{s}_2^\top \bar{A}$ are "far" (w.h.p.)
  - However, efficient "decoding" algorithm to recover $\bar{s}$ from "noisy" $\bar{t}^\top \approx \bar{s}^\top \bar{A}$ *not known*

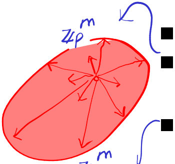


3 Potentially hard: solve "noisy" linear equations over $(\mathbb{Z}_p, +, \cdot)$?

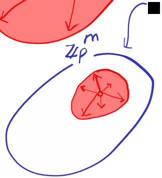
- Input $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}$, $\bar{s} \leftarrow \mathbb{Z}_p^n$, $\bar{e} \leftarrow \mathrm{E}^m$ and

$$\bar{t}^\top := \bar{s}^\top \bar{A} + \bar{e}^\top \bmod p$$





- Solution: $\bar{s}$
- Uninteresting case: E = uniform over $\mathbb{Z}_p$
  - $\bar{t}$ loses information about $\bar{s}$



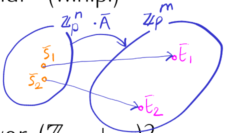
- Interesting: $\mathrm{E} = \mathrm{E}_\alpha$, the *discrete Gaussian distribution* over $\mathbb{Z}$
  - Centred at 0; parameter $\alpha < 1$ determines s.d. $\sigma := \alpha p$

$$\Pr[e] = \frac{1}{\sqrt{2\pi}\sigma \cdot \exp e^2/2\sigma^2}$$

  - $\bar{t}$ "determines" $\bar{s}$, but efficient algorithm to recover $\bar{s}$ not known

Solving Linear Equations Over $(\mathbb{Z}_p, +, \cdot)$...

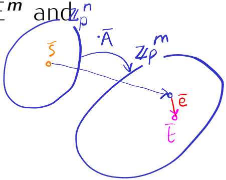
- The map $\bar{s} \mapsto (\bar{s}^\top \bar{A})^\top$ is a "random linear code"
  - Two "codewords" $\bar{t}_1^\top := \bar{s}_1^\top \bar{A}$ and $\bar{t}_2^\top := \bar{s}_2^\top \bar{A}$ are "far" (w.h.p.)
  - However, efficient "decoding" algorithm to recover $\bar{s}$ from "noisy" $\bar{t}^\top \approx \bar{s}^\top \bar{A}$ *not known*

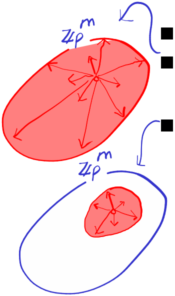


3 Potentially hard: solve "noisy" linear equations over $(\mathbb{Z}_p, +, \cdot)$?

- Input $(\bar{A}, \bar{t})$, where $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}, \bar{s} \leftarrow \mathbb{Z}_p^n, \bar{e} \leftarrow \mathrm{E}^m$ and

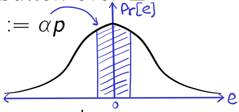
$$\bar{t}^\top := \bar{s}^\top \bar{A} + \bar{e}^\top \bmod p$$





- Solution: $\bar{s}$
- Uninteresting case: E = uniform over $\mathbb{Z}_p$
  - $\bar{t}$ loses information about $\bar{s}$
- Interesting: $\mathrm{E} = \mathrm{E}_\alpha$, the *discrete Gaussian distribution* over $\mathbb{Z}$
  - Centred at 0; parameter $\alpha < 1$ determines s.d. $\sigma := \alpha p$

$$\Pr[e] = \frac{1}{\sqrt{2\pi}\sigma \cdot \exp e^2/2\sigma^2}$$



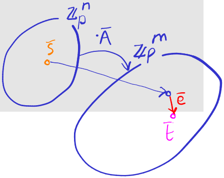
  - $\bar{t}$ "determines" $\bar{s}$, but efficient algorithm to recover $\bar{s}$ not known

Learning with Errors (LWE)

Assumption 1 (Search LWE (SLWE))

The (n, m, p, E)*-SLWE assumption holds if for all quantum polynomial-time (QPT) inverters* Inv *the following is negligible*

$$\Pr_{\substack{\bar{A}\leftarrow \mathbb{Z}_p^{n\times m}\\ \bar{s}\leftarrow\mathbb{Z}_p^{n}, \bar{e}\leftarrow \mathrm{E}^{m}}} \left[\mathsf{Inv}(\bar{A}, \bar{s}^\top \bar{A} + \bar{e}^\top) = \bar{s}\right]$$



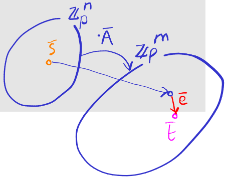
Learning with Errors (LWE)

Assumption 1 (Search LWE (SLWE))

poly(n) E_α, discrete Gaussian

The (n, m, p, E)*-SLWE assumption holds if for all quantum polynomial-time (QPT) inverters* Inv *the following is negligible*

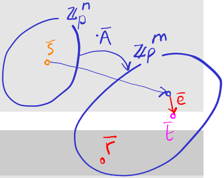
$$\Pr_{\substack{\bar{A} \leftarrow \mathbb{Z}_p^{n \times m} \\ \bar{s} \leftarrow \mathbb{Z}_p^n, \bar{e} \leftarrow \mathrm{E}^m}} \left[\mathrm{Inv}(\bar{A}, \bar{s}^\top \bar{A} + \bar{e}^\top) = \bar{s}\right]$$



Assumption 1 (Search LWE (SLWE))

The (n, m, p, E)*-SLWE assumption holds if for all quantum polynomial-time (QPT) inverters* Inv *the following is negligible*

$$\Pr_{\substack{\bar{A} \leftarrow \mathbb{Z}_p^{n \times m} \\ \bar{s} \leftarrow \mathbb{Z}_p^n, \bar{e} \leftarrow \mathrm{E}^m}} \left[\mathsf{Inv}(\bar{A}, \bar{s}^\top \bar{A} + \bar{e}^\top) = \bar{s}\right]$$



Assumption 2 (Decision LWE (DLWE))

The (n, m, p, E)*-DLWE assumption holds if for all QPT distinguishers* D *the following is negligible*

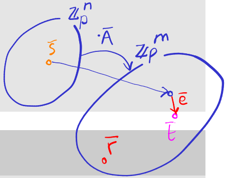
$$\delta(n) := \left| \Pr_{\substack{\bar{A} \leftarrow \mathbb{Z}_p^{n \times m} \\ \bar{s} \leftarrow \mathbb{Z}_p^n, \bar{e} \leftarrow \mathrm{E}^m}} \left[\mathsf{D}(\bar{A}, \bar{s}^\top \bar{A} + \bar{e}^\top) = 0\right] - \Pr_{\substack{\bar{A} \leftarrow \mathbb{Z}_p^{n \times m} \\ \bar{r} \leftarrow \mathbb{Z}_p^m}} \left[\mathsf{D}(\bar{A}, \bar{r}^\top) = 0\right] \right|$$

Learning with Errors (LWE)

Assumption 1 (Search LWE (SLWE))

The (n, m, p, E)*-SLWE assumption holds if for all quantum polynomial-time (QPT) inverters* Inv *the following is negligible*

$$\Pr_{\substack{\bar{A} \leftarrow \mathbb{Z}_p^{n \times m} \\ \bar{s} \leftarrow \mathbb{Z}_p^n, \bar{e} \leftarrow \mathrm{E}^m}} \left[\mathsf{Inv}(\bar{A}, \bar{s}^\top \bar{A} + \bar{e}^\top) = \bar{s} \right]$$



Assumption 2 (Decision LWE (DLWE))

The (n, m, p, E)*-DLWE assumption holds if for all QPT distinguishers* D *the following is negligible*

$$\delta(n) := \left| \Pr_{\substack{\bar{A} \leftarrow \mathbb{Z}_p^{n \times m} \\ \bar{s} \leftarrow \mathbb{Z}_p^n, \bar{e} \leftarrow \mathrm{E}^m}} \left[\mathsf{D}(\bar{A}, \bar{s}^\top \bar{A} + \bar{e}^\top) = 0 \right] - \Pr_{\substack{\bar{A} \leftarrow \mathbb{Z}_p^{n \times m} \\ \bar{r} \leftarrow \mathbb{Z}_p^m}} \left[\mathsf{D}(\bar{A}, \bar{r}^\top) = 0 \right] \right|$$

real world

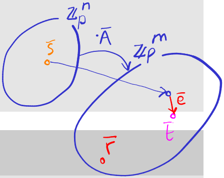
random world

Learning with Errors (LWE)

Assumption 1 (Search LWE (SLWE))

The (n, m, p, E)-SLWE assumption holds if for all quantum polynomial-time (QPT) inverters Inv *the following is negligible*

$$\Pr_{\substack{\bar{A} \leftarrow \mathbb{Z}_p^{n \times m} \\ \bar{s} \leftarrow \mathbb{Z}_p^n, \bar{e} \leftarrow \mathrm{E}^m}} \left[\mathsf{Inv}(\bar{A}, \bar{s}^\top \bar{A} + \bar{e}^\top) = \bar{s}\right]$$



Assumption 2 (Decision LWE (DLWE))

The (n, m, p, E)-DLWE assumption holds if for all QPT distinguishers D *the following is negligible*

$$\delta(n) := \left| \Pr_{\substack{\bar{A} \leftarrow \mathbb{Z}_p^{n \times m} \\ \bar{s} \leftarrow \mathbb{Z}_p^n, \bar{e} \leftarrow \mathrm{E}^m}} \left[\mathsf{D}(\bar{A}, \bar{s}^\top \bar{A} + \bar{e}^\top) = 0\right] - \Pr_{\substack{\bar{A} \leftarrow \mathbb{Z}_p^{n \times m} \\ \bar{r} \leftarrow \mathbb{Z}_p^m}} \left[\mathsf{D}(\bar{A}, \bar{r}^\top) = 0\right] \right|$$

real world   random world

Exercise 1

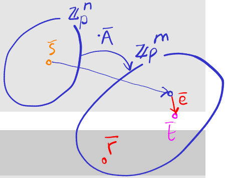
Are DLWE and SLWE random self-reducible?

Learning with Errors (LWE)

Assumption 1 (Search LWE (SLWE))

The (n, m, p, E)-SLWE assumption holds if for all quantum polynomial-time (QPT) inverters Inv *the following is negligible*

$$\Pr_{\substack{\bar{A} \leftarrow \mathbb{Z}_p^{n\times m} \\ \bar{s} \leftarrow \mathbb{Z}_p^n, \bar{e} \leftarrow \mathrm{E}^m}} \left[\mathsf{Inv}(\bar{A}, \bar{s}^\top \bar{A} + \bar{e}^\top) = \bar{s}\right]$$



Assumption 2 (Decision LWE (DLWE))

The (n, m, p, E)-DLWE assumption holds if for all QPT distinguishers D *the following is negligible*

$$\delta(n) := \Pr_{\substack{\bar{A} \leftarrow \mathbb{Z}_p^{n\times m} \\ \bar{s} \leftarrow \mathbb{Z}_p^n, \bar{e} \leftarrow \mathrm{E}^m}} \left[\mathsf{D}(\bar{A}, \bar{s}^\top \bar{A} + \bar{e}^\top) = 0\right] - \Pr_{\substack{\bar{A} \leftarrow \mathbb{Z}_p^{n\times m} \\ \bar{r} \leftarrow \mathbb{Z}_p^m}} \left[\mathsf{D}(\bar{A}, \bar{r}^\top) = 0\right]$$

real world

random world

Exercise 1

Are DLWE and SLWE random self-reducible?

Decision and Search LWE are Equivalent!

- Note: this is not true for, e.g, CDH and DDH!

Decision and Search LWE are Equivalent!

- Note: this is not true for, e.g, CDH and DDH!

Claim 1 (Search to decision reduction for LWE)

For any $n \in \mathbb{N}$, m, $p \in \mathsf{poly}(n)$ and E, and sufficiently large m', (n, m', p, E)-SLWE problem reduces to (n, m, p, E)-DLWE problem.

Decision and Search LWE are Equivalent!

- Note: this is not true for, e.g, CDH and DDH!

Claim 1 (Search to decision reduction for LWE)

For any $n \in \mathbb{N}$, m, $p \in \mathsf{poly}(n)$ *and* E, *and sufficiently large* m', (n, m', p, E)*-SLWE problem reduces to* (n, m, p, E)*-DLWE problem.*

Proof sketch. $\exists \mathsf{Inv}$ for SLWE $\Leftarrow \exists \mathsf{D}$ for DLWE.

- Assume *perfect* dist. for single sample $(\bar{a}, \bar{s}^\top \bar{a} + e)$ and $(\bar{a}, r)$
- Focus on extracting first coordinate s_1 of $\bar{s}$

Decision and Search LWE are Equivalent!

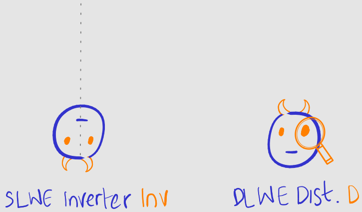
- Note: this is not true for, e.g, CDH and DDH!

Claim 1 (Search to decision reduction for LWE)

For any $n \in \mathbb{N}$, m, $p \in \mathsf{poly}(n)$ and E, and sufficiently large m', (n, m', p, E)-SLWE problem reduces to (n, m, p, E)-DLWE problem.

Proof sketch. $\exists$Inv for SLWE $\Leftarrow$ $\exists$D for DLWE.

- Assume *perfect* dist. for single sample $(\bar{a}, \bar{s}^\top \bar{a} + e)$ and $(\bar{a}, r)$
- Focus on extracting first coordinate s_1 of $\bar{s}$



Decision and Search LWE are Equivalent!

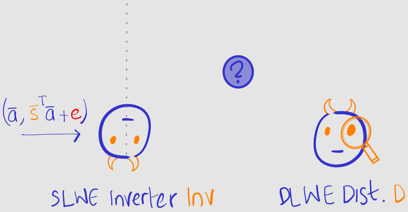
- Note: this is not true for, e.g, CDH and DDH!

Claim 1 (Search to decision reduction for LWE)

For any $n \in \mathbb{N}$, m, $p \in \mathsf{poly}(n)$ and E, and sufficiently large m', (n, m', p, E)-SLWE problem reduces to (n, m, p, E)-DLWE problem.

Proof sketch. $\exists$Inv for SLWE $\Leftarrow$ $\exists$D for DLWE.

- Assume *perfect* dist. for single sample $(\bar{a}, \bar{s}^\top \bar{a} + e)$ and $(\bar{a}, r)$
- Focus on extracting first coordinate s_1 of $\bar{s}$



Decision and Search LWE are Equivalent!

- Note: this is not true for, e.g, CDH and DDH!

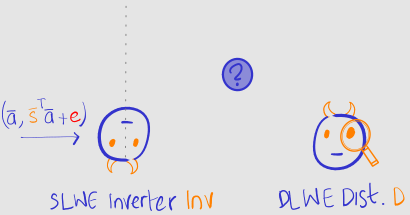
Claim 1 (Search to decision reduction for LWE)

For any $n \in \mathbb{N}$, $m, p \in \mathsf{poly}(n)$ and E, and sufficiently large m', (n, m', p, E)-SLWE problem reduces to (n, m, p, E)-DLWE problem.

Proof sketch. $\exists \mathsf{Inv}$ for SLWE $\Leftarrow \exists \mathsf{D}$ for DLWE.

- Assume *perfect* dist. for single sample $(\bar{a}, \bar{s}^\top \bar{a} + e)$ and $(\bar{a}, r)$
- Focus on extracting first coordinate s_1 of $\bar{s}$

◆ Is it possible to transform $(\bar{a}, \bar{s}^T\bar{a}+e)$ into another sample $(\bar{a}', \bar{s}^T\bar{a}'+e')$?



Decision and Search LWE are Equivalent!

- Note: this is not true for, e.g, CDH and DDH!

Claim 1 (Search to decision reduction for LWE)

For any $n \in \mathbb{N}$, $m, p \in \mathsf{poly}(n)$ *and* E, *and sufficiently large* m', (n, m', p, E)*-SLWE problem reduces to* (n, m, p, E)*-DLWE problem.*

Proof sketch. $\exists$Inv for SLWE $\Leftarrow$ $\exists$D for DLWE.

- Assume *perfect* dist. for single sample $(\bar{a}, \bar{s}^\top \bar{a} + e)$ and $(\bar{a}, r)$
- Focus on extracting first coordinate s_1 of $\bar{s}$

◆ Is it possible to transform $(\bar{a}, \bar{s}^\top\bar{a}+e)$ into another sample $(\bar{a}', \bar{s}^\top\bar{a}'+e')$?

◆ What if you knew s_1?

Decision and Search LWE are Equivalent!

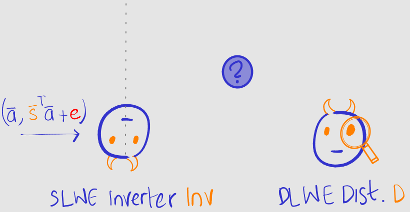
- Note: this is not true for, e.g, CDH and DDH!

Claim 1 (Search to decision reduction for LWE)

For any $n \in \mathbb{N}$, $m, p \in \mathsf{poly}(n)$ and E, and sufficiently large m', (n, m', p, E)-SLWE problem reduces to (n, m, p, E)-DLWE problem.

Proof sketch. $\exists$Inv for SLWE $\Leftarrow$ $\exists$D for DLWE.

- Assume *perfect* dist. for single sample $(\bar{a}, \bar{s}^\top \bar{a} + e)$ and $(\bar{a}, r)$
- Focus on extracting first coordinate s_1 of $\bar{s}$



◆ Is it possible to transform $(\bar{a}, \bar{s}^\top \bar{a} + e)$ into another sample $(\bar{a}', \bar{s}^\top \bar{a}' + e)$?

◆ What if you knew s_1? Sample $a_1' \leftarrow \mathbb{Z}_p$:

$$(\bar{a}, \bar{s}^\top \bar{a} + e) \mapsto (\bar{a} + (a_1', 0, \ldots, 0), \bar{s}^\top \bar{a} + e + a_1' s_1)$$
$$= (\bar{a}', \bar{s}^\top \bar{a}' + e)$$

Decision and Search LWE are Equivalent!

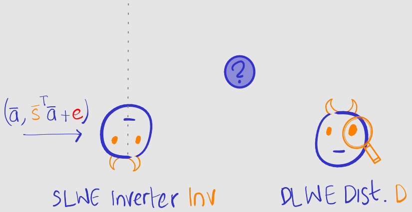
- Note: this is not true for, e.g, CDH and DDH!

Claim 1 (Search to decision reduction for LWE)

For any $n \in \mathbb{N}$, $m, p \in \mathsf{poly}(n)$ *and* E, *and sufficiently large* m', (n, m', p, E)*-SLWE problem reduces to* (n, m, p, E)*-DLWE problem.*

Proof sketch. $\exists\mathsf{Inv}$ for SLWE $\Leftarrow$ $\exists\mathsf{D}$ for DLWE.

- Assume *perfect* dist. for single sample $(\bar{a}, \bar{s}^\top \bar{a} + e)$ and $(\bar{a}, r)$
- Focus on extracting first coordinate s_1 of $\bar{s}$



◆ Is it possible to transform $(\bar{a}, \bar{s}^\top \bar{a} + e)$ into another sample $(\bar{a}', \bar{s}^\top \bar{a}' + e)$?

◆ What if you knew s_1? Sample $a'_1 \leftarrow \mathbb{Z}_p$:

$$(\bar{a}, \bar{s}^\top \bar{a} + e) \mapsto (\bar{a} + (a'_1, 0, \ldots, 0), \bar{s}^\top \bar{a} + e + a'_1 s_1) = (\bar{a}', \bar{s}^\top \bar{a}' + e)$$

◆ Why not guess s_1?

Decision and Search LWE are Equivalent!

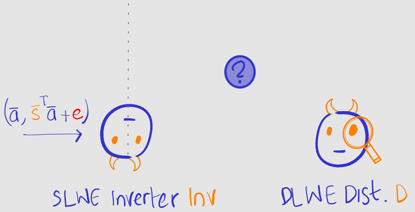
- Note: this is not true for, e.g, CDH and DDH!

Claim 1 (Search to decision reduction for LWE)

For any $n \in \mathbb{N}$, $m, p \in \text{poly}(n)$ and E, and sufficiently large m', (n, m', p, E)-SLWE problem reduces to (n, m, p, E)-DLWE problem.

Proof sketch. $\exists$Inv for SLWE $\Leftarrow$ $\exists$D for DLWE.

- Assume *perfect* dist. for single sample $(\bar{a}, \bar{s}^\top \bar{a} + e)$ and $(\bar{a}, r)$
- Focus on extracting first coordinate s_1 of $\bar{s}$



- Is it possible to transform $(\bar{a}, \bar{s}^T\bar{a}+e)$ into another sample $(\bar{a}', \bar{s}^T\bar{a}'+e')$?
- What if you knew s_1? Sample $a_1' \leftarrow \mathbb{Z}_p$:
  $(\bar{a}, \bar{s}^T\bar{a}+e) \mapsto (\bar{a} + (a_1', 0, \ldots, 0), \bar{s}^T\bar{a} + e + a_1's_1)$
  $= (\bar{a}', \bar{s}^T\bar{a}'+e)$
- Why not guess s_1? What if guess wrong?
  $(\bar{a}, \bar{s}^T\bar{a}+e) \mapsto (\bar{a}', \bar{s}^T\bar{a} + e + a_1's_1)$ random!

Decision and Search LWE are Equivalent!

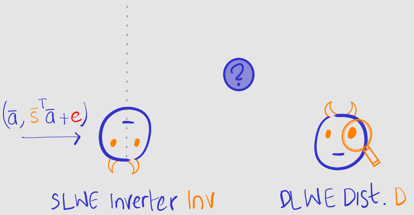
- Note: this is not true for, e.g, CDH and DDH!

Claim 1 (Search to decision reduction for LWE)

For any $n \in \mathbb{N}$, m, $p \in \mathsf{poly}(n)$ and E, and sufficiently large m', (n, m', p, E)-SLWE problem reduces to (n, m, p, E)-DLWE problem.

Proof sketch. $\exists \mathsf{Inv}$ for SLWE $\Leftarrow \exists \mathsf{D}$ for DLWE.

- Assume *perfect* dist. for single sample $(\bar{a}, \bar{s}^\top \bar{a} + e)$ and $(\bar{a}, r)$
- Focus on extracting first coordinate s_1 of $\bar{s}$



- Is it possible to transform $(\bar{a}, \bar{s}^T\bar{a}+e)$ into another sample $(\bar{a}', \bar{s}^T\bar{a}'+e)$?
- What if you knew s_1? Sample $a_1' \leftarrow \mathbb{Z}_p$:
  $(\bar{a}, \bar{s}^T\bar{a}+e) \mapsto (\bar{a}+(a_1', 0, \ldots, 0), \bar{s}^T\bar{a}+e+a_1's_1)$
  $= (\bar{a}', \bar{s}^T\bar{a}'+e)$
- Why not guess s_1? What if guess wrong?
  $(\bar{a}, \bar{s}^T\bar{a}+e) \mapsto (\bar{a}', \bar{s}^T\bar{a}+e+a_1's_1)$ random!
- Iterate over all possible s_1

Decision and Search LWE are Equivalent!

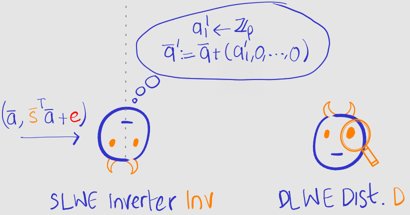
- Note: this is not true for, e.g, CDH and DDH!

Claim 1 (Search to decision reduction for LWE)

For any $n \in \mathbb{N}$, $m, p \in \mathsf{poly}(n)$ and E, and sufficiently large m', (n, m', p, E)-SLWE problem reduces to (n, m, p, E)-DLWE problem.

Proof sketch. $\exists \mathsf{Inv}$ for SLWE $\Leftarrow \exists \mathsf{D}$ for DLWE.

- Assume *perfect* dist. for single sample $(\bar{a}, \bar{s}^\top \bar{a} + e)$ and $(\bar{a}, r)$
- Focus on extracting first coordinate s_1 of $\bar{s}$



- Is it possible to transform $(\bar{a}, \bar{s}^T \bar{a} + e)$ into another sample $(\bar{a}', \bar{s}^T \bar{a}' + e)$?
- What if you knew s_1? Sample $a_1' \leftarrow \mathbb{Z}_p$:

$$(\bar{a}, \bar{s}^T \bar{a} + e) \mapsto (\bar{a} + (a_1', 0, \dots, 0), \bar{s}^T \bar{a} + e + a_1' s_1) = (\bar{a}', \bar{s}^T \bar{a}' + e)$$

- Why not guess s_1? What if guess wrong?

$$(\bar{a}, \bar{s}^T \bar{a} + e) \mapsto (\bar{a}', \bar{s}^T \bar{a} + e + a_1' s_1) \text{ random!}$$

- Iterate over all possible s_1

Decision and Search LWE are Equivalent!

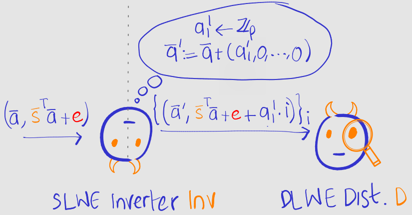
- Note: this is not true for, e.g, CDH and DDH!

Claim 1 (Search to decision reduction for LWE)

For any $n \in \mathbb{N}$, $m, p \in \mathsf{poly}(n)$ and E, and sufficiently large m', (n, m', p, E)-SLWE problem reduces to (n, m, p, E)-DLWE problem.

Proof sketch. $\exists\mathsf{Inv}$ for SLWE $\Leftarrow \exists\mathsf{D}$ for DLWE.

- Assume *perfect* dist. for single sample $(\bar{a}, \bar{s}^\top \bar{a} + e)$ and $(\bar{a}, r)$
- Focus on extracting first coordinate s_1 of $\bar{s}$



◆ Is it possible to transform $(\bar{a}, \bar{s}^T\bar{a} + e)$ into another sample $(\bar{a}', \bar{s}^T\bar{a}' + e')$?

◆ What if you knew s_1? Sample $a_1 \leftarrow \mathbb{Z}_p$:

$$(\bar{a}, \bar{s}^T\bar{a} + e) \longmapsto (\bar{a} + (a_1', 0, \dots, 0), \bar{s}^T\bar{a} + e + a_1' s_1)$$
$$= (\bar{a}', \bar{s}^T\bar{a}' + e)$$

◆ Why not guess s_1? What if guess wrong?

$$(\bar{a}, \bar{s}^T\bar{a} + e) \longmapsto (\bar{a}', \bar{s}^T\bar{a} + e + a_1' s_1) \text{ random!}$$

◆ Iterate over all possible s_1

Decision and Search LWE are Equivalent!

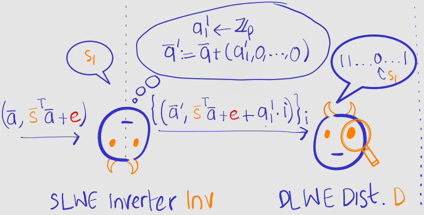
- Note: this is not true for, e.g, CDH and DDH!

Claim 1 (Search to decision reduction for LWE)

For any $n \in \mathbb{N}$, $m, p \in \mathrm{poly}(n)$ and E, and sufficiently large m', (n, m', p, E)-SLWE problem reduces to (n, m, p, E)-DLWE problem.

Proof sketch. $\exists$Inv for SLWE $\Leftarrow \exists$D for DLWE.

- Assume *perfect* dist. for single sample $(\bar{a}, \bar{s}^\top \bar{a} + e)$ and $(\bar{a}, r)$
- Focus on extracting first coordinate s_1 of $\bar{s}$



- Is it possible to transform $(\bar{a}, \bar{s}^T\bar{a}+e)$ into another sample $(\bar{a}', \bar{s}^T\bar{a}'+e')$?
- What if you knew s_1? Sample $a_1' \leftarrow \mathbb{Z}_p$:
  $(\bar{a}, \bar{s}^T\bar{a}+e) \mapsto (\bar{a}+(a_1', 0, \dots, 0), \bar{s}^T\bar{a}+e+a_1's_1)$
  $= (\bar{a}', \bar{s}^T\bar{a}'+e)$
- Why not guess s_1? What if guess wrong?
  $(\bar{a}, \bar{s}^T\bar{a}+e) \mapsto (\bar{a}', \bar{s}^T\bar{a}+e+a_1's_1)$ random!
- Iterate over all possible s_1

Decision and Search LWE are Equivalent!

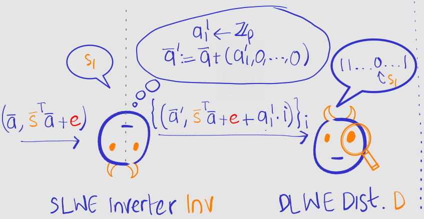
- Note: this is not true for, e.g, CDH and DDH!

Claim 1 (Search to decision reduction for LWE)

For any $n \in \mathbb{N}$, $m, p \in \mathsf{poly}(n)$ and E, and sufficiently large m', (n, m', p, E)-SLWE problem reduces to (n, m, p, E)-DLWE problem.

Proof sketch. $\exists \mathsf{Inv}$ for SLWE $\Leftarrow \exists \mathsf{D}$ for DLWE.

- Assume *perfect* dist. for single sample $(\bar{a}, \bar{s}^\top \bar{a} + e)$ and $(\bar{a}, r)$
- Focus on extracting first coordinate s_1 of $\bar{s}$



- ◆ Is it possible to transform $(\bar{a}, \bar{s}^T \bar{a} + e)$ into another sample $(\bar{a}', \bar{s}^T \bar{a}' + e')$?
- ◆ What if you knew s_1? Sample $a_1' \leftarrow \mathbb{Z}_p$:

  $(\bar{a}, \bar{s}^T \bar{a} + e) \mapsto (\bar{a} + (a_1', 0, \dots, 0), \bar{s}^T \bar{a} + e + a_1' s_1)$
  $= (\bar{a}', \bar{s}^T \bar{a}' + e)$
- ◆ Why not guess s_1? What if guess wrong?

  $(\bar{a}, \bar{s}^T \bar{a} + e) \mapsto (\bar{a}', \bar{s}^T \bar{a} + e + a_1' s_1)$ random!
- ◆ Iterate over all possible s_1

$\square$

Plan for this Lecture

1 Motivation: Quantum Adversaries

2 Learning with Errors (LWE)

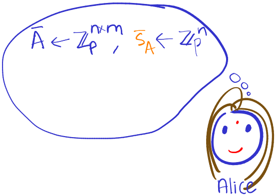
3 Cryptography from LWE

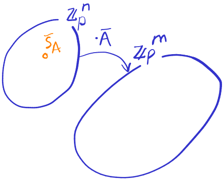
4 LWE and Lattices

1-Bit Key-Exchange Protocol ← DLWE...

- The protocol:

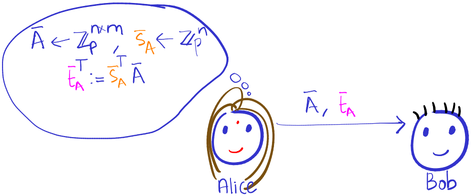
1-Bit Key-Exchange Protocol ← DLWE...

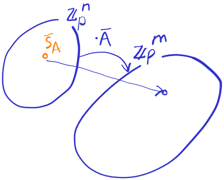




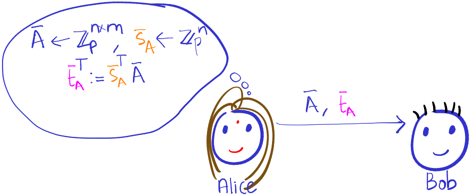
- The protocol:

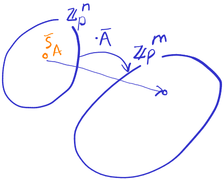
1-Bit Key-Exchange Protocol ← DLWE...



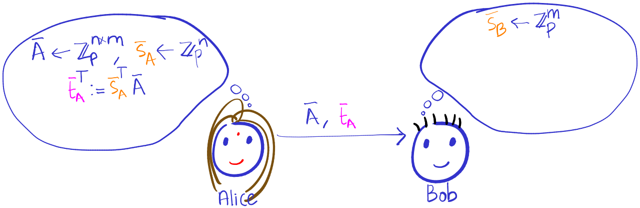


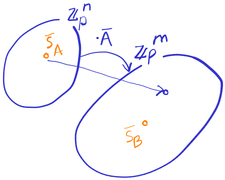
- The protocol:





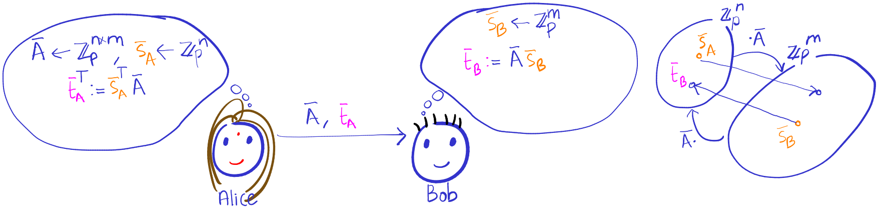
- The protocol:
  1. Alice→Bob: send $(\bar{A}, \bar{t}_A^\top := \bar{s}_A^\top \bar{A})$, where
     - $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}$, $\bar{s}_A \leftarrow \mathbb{Z}_p^n$



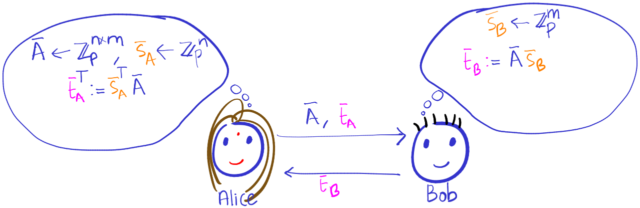


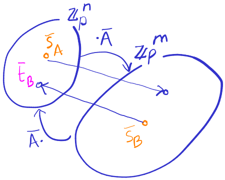
- The protocol:
  1. Alice→Bob: send ($\bar{A}, \bar{t}_A^\top := \bar{s}_A^\top \bar{A}$), where
     - $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}$, $\bar{s}_A \leftarrow \mathbb{Z}_p^n$

1-Bit Key-Exchange Protocol ← DLWE

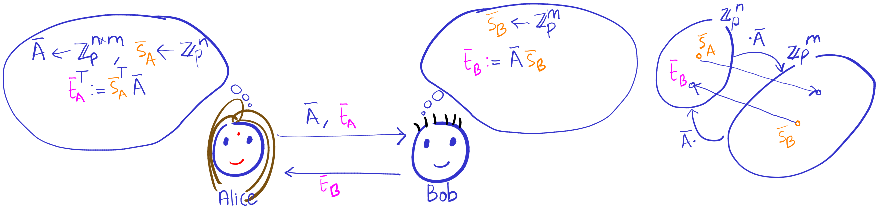


- The protocol:
  1. Alice→Bob: send $(\bar{A}, \bar{t}_A^\top := \bar{s}_A^\top \bar{A})$, where
     - $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}$, $\bar{s}_A \leftarrow \mathbb{Z}_p^n$

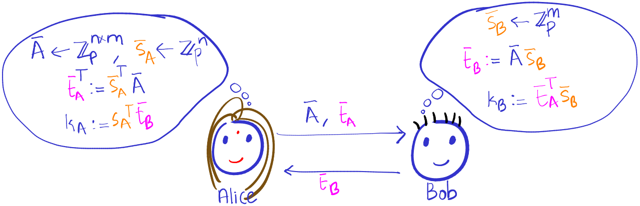


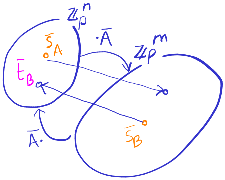


- The protocol:
  1. Alice→Bob: send $(\bar{A}, \bar{t}_A^\top := \bar{s}_A^\top \bar{A})$, where
     - $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}$, $\bar{s}_A \leftarrow \mathbb{Z}_p^n$

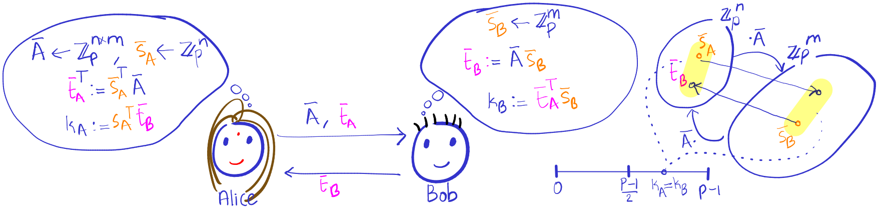


- The protocol:
  1. Alice→Bob: send $(\bar{A}, \bar{t}_A^\top := \bar{s}_A^\top \bar{A} + \bar{e}^\top)$, where
     - $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}$, $\bar{s}_A \leftarrow \mathbb{Z}_p^n$
  2. Alice←Bob: send $(\bar{t}_B := \bar{A}\bar{s}_B \quad |)$, where
     - $\bar{s}_B \leftarrow \mathbb{Z}_p^m$

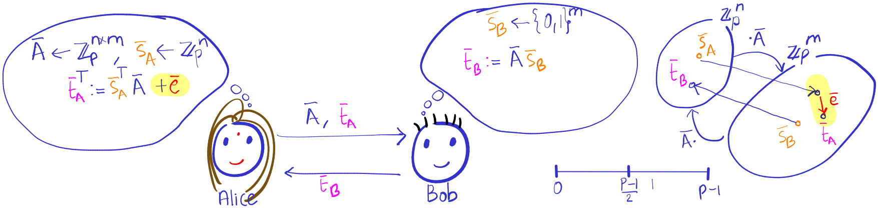




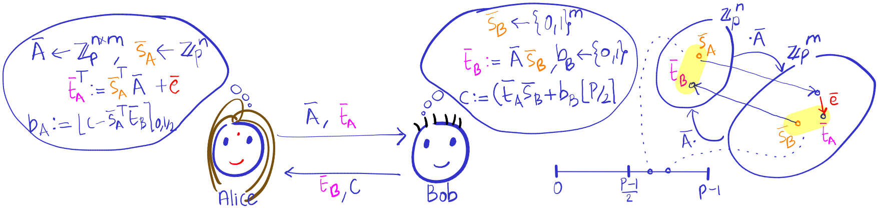
- The protocol:
  1. Alice→Bob: send $(\bar{A}, \bar{t}_A^\top := \bar{s}_A^\top \bar{A} + \bar{e}^\top)$, where
     - $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}$, $\bar{s}_A \leftarrow \mathbb{Z}_p^n$
  2. Alice←Bob: send $(\bar{t}_B := \bar{A}\bar{s}_B$ $)$, where
     - $\bar{s}_B \leftarrow \mathbb{Z}_p^m$



- The protocol:
  1. Alice→Bob: send $(\bar{A}, \bar{t}_A^\top := \bar{s}_A^\top \bar{A} + \bar{e}^\top)$, where
     - $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}$, $\bar{s}_A \leftarrow \mathbb{Z}_p^n$
  2. Alice←Bob: send $(\bar{t}_B := \bar{A}\bar{s}_B$ $)$, where
     - $\bar{s}_B \leftarrow \mathbb{Z}_p^m$

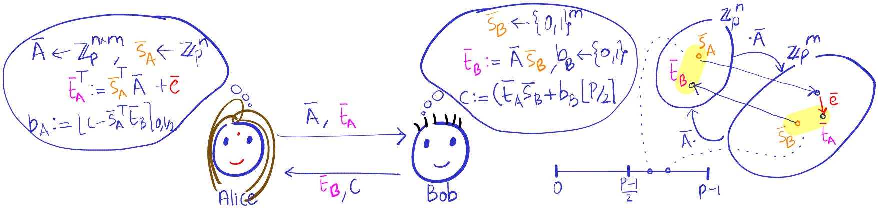


- The protocol:
  1. Alice→Bob: send $(\bar{A}, \bar{t}_A^\top := \bar{s}_A^\top \bar{A} + \bar{e}^\top)$, where
     - $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}$, $\bar{s}_A \leftarrow \mathbb{Z}_p^n$
     - $\bar{e} \leftarrow \mathrm{E}_\alpha^m$
  2. Alice←Bob: send $(\bar{t}_B := \bar{A}\bar{s}_B)$, where
     - $\bar{s}_B \leftarrow \{0,1\}^m$



- The protocol:
  1. Alice→Bob: send $(\bar{A}, \bar{t}_A^\top := \bar{s}_A^\top \bar{A} + \bar{e}^\top)$, where
     - $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}$, $\bar{s}_A \leftarrow \mathbb{Z}_p^n$
     - $\bar{e} \leftarrow \mathrm{E}_\alpha^m$
  2. Alice←Bob: send $(\bar{t}_B := \bar{A}\bar{s}_B, c := (\bar{t}_A^\top \bar{s}_B + b_B \lfloor p/2 \rceil)$), where
     - $\bar{s}_B \leftarrow \{0,1\}^m$
     - $b_B \leftarrow \{0,1\}$
  3. Alice outputs $b_A := \lfloor c - \bar{s}_A^\top \bar{t}_B \rceil_{0,1/2}$ and Bob outputs b_B

1-Bit Key-Exchange Protocol $\leftarrow$ DLWE...

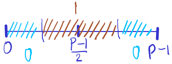


- The protocol:

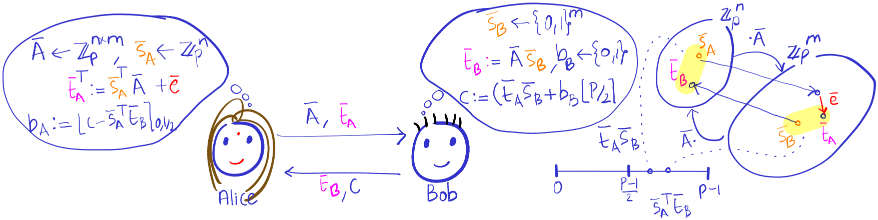
  1. Alice$\rightarrow$Bob: send $(\bar{A}, \bar{t}_A^\top := \bar{s}_A^\top \bar{A} + \bar{e}^\top)$, where
     - $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}$, $\bar{s}_A \leftarrow \mathbb{Z}_p^n$
     - $\bar{e} \leftarrow E_\alpha^m$

     round

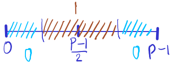
  2. Alice$\leftarrow$Bob: send $(\bar{t}_B := \bar{A}\bar{s}_B, c := (\bar{t}_A^\top \bar{s}_B + b_B \lfloor p/2 \rceil))$, where
     - $\bar{s}_B \leftarrow \{0,1\}^m$
     - $b_B \leftarrow \{0,1\}$



  3. Alice outputs $b_A := \lfloor c - \bar{s}_A^\top \bar{t}_B \rceil_{0,1/2}$ and Bob outputs b_B

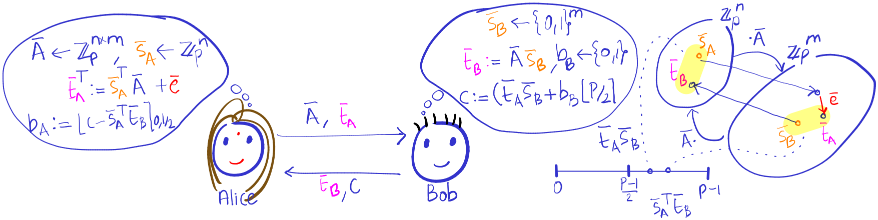


- The protocol:
  1. Alice→Bob: send $(\bar{A}, \bar{t}_A^\top := \bar{s}_A^\top \bar{A} + \bar{e}^\top)$, where
     - $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}$, $\bar{s}_A \leftarrow \mathbb{Z}_p^n$
     - $\bar{e} \leftarrow \mathrm{E}_\alpha^m$
  2. Alice←Bob: send $(\bar{t}_B := \bar{A}\bar{s}_B, c := (\bar{t}_A^\top \bar{s}_B + b_B \lfloor p/2 \rceil)$, where round
     - $\bar{s}_B \leftarrow \{0,1\}^m$
     - $b_B \leftarrow \{0,1\}$



  3. Alice outputs $b_A := \lfloor c - \bar{s}_A^\top \bar{t}_B \rceil_{0,1/2}$ and Bob outputs b_B

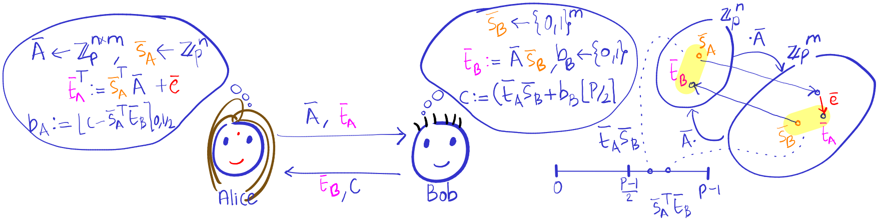
1-Bit Key-Exchange Protocol ← DLWE...



- Correctness of key generation:

- Scheme has negligible *key-exchange error* if $\alpha \leq 1/\tilde{O}(\sqrt{n})$

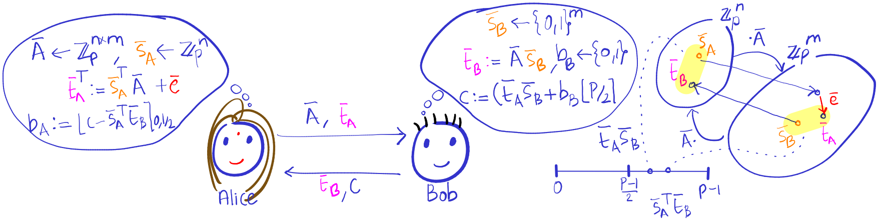
1-Bit Key-Exchange Protocol ← DLWE...



- Correctness of key generation:

  Note that $c - \bar{s}_A^T \bar{t}_B = \bar{t}_A^T \cdot \bar{s}_B - \bar{s}_A^T \bar{t}_B + b_A \lfloor p/2 \rceil$

- Scheme has negligible *key-exchange error* if $\alpha \leq 1/\tilde{O}(\sqrt{n})$

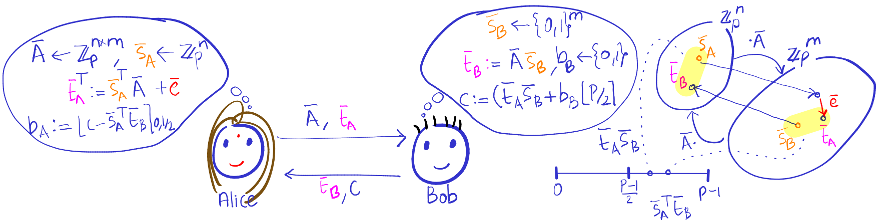


- Correctness of key generation:

Note that $c - \bar{s}_A^T \bar{t}_B = \bar{t}_A^T \cdot \bar{s}_B - \bar{s}_A^T \bar{t}_B + b_A \lceil p/2 \rceil$

$$= (\bar{s}_A^T \bar{A} + e^T)\, \bar{s}_B - \bar{s}_A^T \bar{A} s_B + b_A \lceil p/2 \rceil$$

- Scheme has negligible *key-exchange error* if $\alpha \leq 1/\tilde{O}(\sqrt{n})$

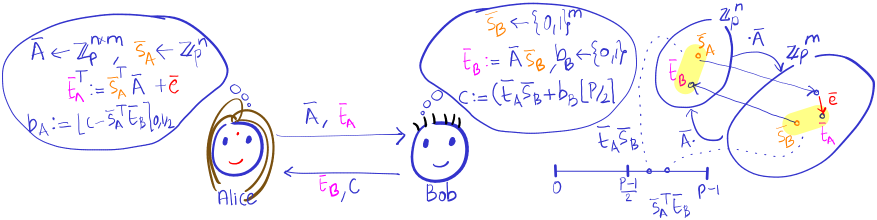


- Correctness of key generation:

Note that
$$c - \bar{s}_A^T \bar{t}_B = \bar{t}_A^T \cdot \bar{s}_B - \bar{s}_A^T \bar{t}_B + b_A \lfloor p/2 \rceil$$
$$= (\cancel{\bar{s}_A^T \bar{A}} + e^T)\,\bar{s}_B - \cancel{\bar{s}_A^T \bar{A} \bar{s}_B} + b_A \lfloor p/2 \rceil$$
$$= e^T \bar{s}_B + b_A \lfloor p/2 \rceil$$

- Scheme has negligible *key-exchange error* if $\alpha \leq 1/\tilde{O}(\sqrt{n})$

1-Bit Key-Exchange Protocol ← DLWE...



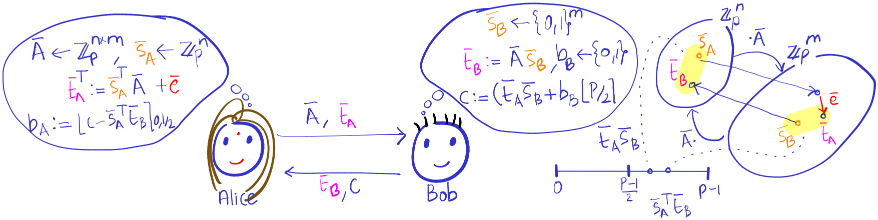
- Correctness of key generation:

Note that $c - \bar{s}_A^T \bar{t}_B = \bar{t}_A^T \cdot \bar{s}_B - \bar{s}_A^T \bar{t}_B + b_A \lfloor p/2 \rceil$

$$= (\cancel{\bar{s}_A^T \bar{A}} + e^T)\, \bar{s}_B - \cancel{\bar{s}_A^T \bar{A} \bar{s}_B} + b_A \lfloor p/2 \rceil$$

$$= e^T \bar{s}_B + b_A \lfloor p/2 \rceil \implies b_A = b_B \text{ if } |e^T \bar{s}_B| < p/4$$

- Scheme has negligible *key-exchange error* if $\alpha \leq 1/\tilde{O}(\sqrt{n})$



- Correctness of key generation:

Note that $c - \bar{s}_A^T \bar{t}_B = \bar{t}_A^T \cdot \bar{s}_B - \bar{s}_A^T \bar{t}_B + b_A \lfloor p/2 \rceil$

$$= (\cancel{\bar{s}_A^T \bar{A}} + e^T)\,\bar{s}_B - \cancel{\bar{s}_A^T \bar{A} \bar{s}_B} + b_A \lfloor p/2 \rceil$$

$$= e^T \bar{s}_B + b_A \lfloor p/2 \rceil \implies b_A = b_B \text{ if } |e^T \bar{s}_B| < p/4$$

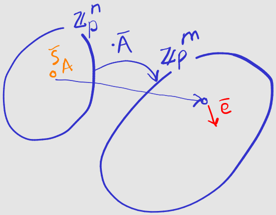
$$\Uparrow$$

- Scheme has negligible *key-exchange error* if $\alpha \leq 1/\tilde{O}(\sqrt{n})$

Regev's Encryption: 1-Bit PKE ← DLWE

Construction 1

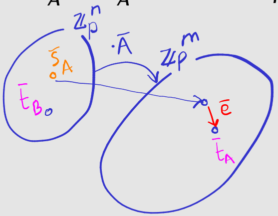
- *Key generation* $\mathsf{Gen}(1^n)$*:*
  1. *Sample matrix* $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}$ *for* $m, p = \mathrm{poly}(n)$
  2. *Sample secret key* $\bar{s}_A \leftarrow \mathbb{Z}_p^n$ *and error* $\bar{e} \leftarrow E_\alpha^m$



Regev's Encryption: 1-Bit PKE ← DLWE

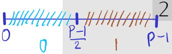
Construction 1

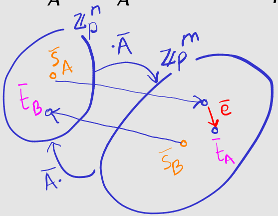
- *Key generation* $\mathsf{Gen}(1^n)$*:*
  1. *Sample matrix* $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}$ *for* $m, p = \mathrm{poly}(n)$
  2. *Sample secret key* $\bar{s}_A \leftarrow \mathbb{Z}_p^n$ *and error* $\bar{e} \leftarrow \mathrm{E}_\alpha^m$
  3. *Output* $\left(\mathrm{pk} := \begin{pmatrix} \bar{A} \\ \bar{t}_A^\top \end{pmatrix}, \mathrm{sk} := \bar{s}_A\right)$, *where* $\bar{t}_A^\top := \bar{s}_A^\top \bar{A} + \bar{e}^\top \bmod p$



Construction 1

- *Key generation* $\mathsf{Gen}(1^n)$*:*
  1. *Sample matrix* $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}$ *for* $m, p = \mathrm{poly}(n)$
  2. *Sample secret key* $\bar{s}_A \leftarrow \mathbb{Z}_p^n$ *and error* $\bar{e} \leftarrow \mathrm{E}_\alpha^m$
  3. *Output* $\left(\mathrm{pk} := \begin{pmatrix} \bar{A} \\ \bar{t}_A^\top \end{pmatrix}, \mathrm{sk} := \bar{s}_A\right)$*, where* $\bar{t}_A^\top := \bar{s}_A^\top \bar{A} + \bar{e}^\top \bmod p$
- *Encryption* $\mathsf{Enc}(\mathrm{pk}, b)$*:*
  1. *Sample random coin* $\bar{s}_B \leftarrow \{0,1\}^m$
  2. *Encode message* $\tilde{b} := b \cdot \lfloor p/2 \rceil$

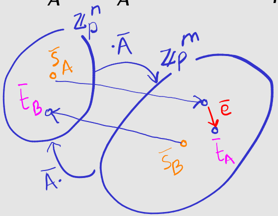




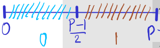
Regev's Encryption: 1-Bit PKE $\leftarrow$ DLWE

Construction 1

- *Key generation* $\mathsf{Gen}(1^n)$:
  1. *Sample matrix* $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}$ *for* $m, p = \mathrm{poly}(n)$
  2. *Sample secret key* $\bar{s}_A \leftarrow \mathbb{Z}_p^n$ *and error* $\bar{e} \leftarrow \mathrm{E}_\alpha^m$
  3. *Output* $(\mathrm{pk} := \begin{pmatrix} \bar{A} \\ \bar{t}_A^\top \end{pmatrix}, \mathrm{sk} := \bar{s}_A)$, *where* $\bar{t}_A^\top := \bar{s}_A^\top \bar{A} + \bar{e}^\top \bmod p$



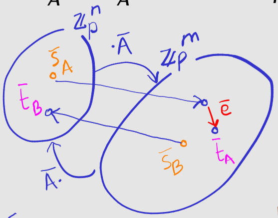
- *Encryption* $\mathsf{Enc}(\mathrm{pk}, b)$:
  1. *Sample random coin* $\bar{s}_B \leftarrow \{0,1\}^m$
  2. *Encode message* $\tilde{b} := b \cdot \lfloor p/2 \rceil$
  3. *Output* $\bar{c} := \mathrm{pk}\bar{s}_B + \begin{pmatrix} 0^n \\ \tilde{b} \end{pmatrix} \bmod p$



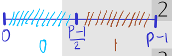
$$\begin{pmatrix} \bar{A}\bar{s}_B \\ \bar{t}_A^\top \bar{s}_B \end{pmatrix}$$

Construction 1

- *Key generation* $\mathsf{Gen}(1^n)$:
  1. *Sample matrix* $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}$ *for* $m, p = \mathrm{poly}(n)$
  2. *Sample secret key* $\bar{s}_A \leftarrow \mathbb{Z}_p^n$ *and error* $\bar{e} \leftarrow \mathrm{E}_\alpha^m$
  3. *Output* $\left(\mathrm{pk} := \begin{pmatrix} \bar{A} \\ \bar{t}_A^\top \end{pmatrix}, \mathrm{sk} := \bar{s}_A\right)$, *where* $\bar{t}_A^\top := \bar{s}_A^\top \bar{A} + \bar{e}^\top \bmod p$



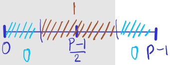
- *Encryption* $\mathsf{Enc}(\mathrm{pk}, b)$:
  1. *Sample random coin* $\bar{s}_B \leftarrow \{0,1\}^m$
  2. *Encode message* $\tilde{b} := b \cdot \lfloor p/2 \rceil$
  3. *Output* $\bar{c} := \mathrm{pk}\bar{s}_B + \begin{pmatrix} 0^n \\ \tilde{b} \end{pmatrix} \bmod p$



$\begin{pmatrix} \bar{A}\bar{s}_B \\ \bar{t}_A^\top \bar{s}_B \end{pmatrix}$

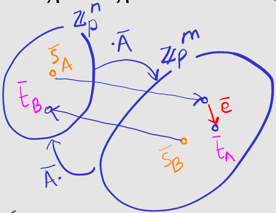
$-\bar{s}_A^\top \bar{A}\bar{s}_B + \bar{t}_A^\top \bar{s}_B$

- *Decryption* $\mathsf{Dec}(\mathrm{sk}, \bar{c})$: *output* $\left\lfloor (-\bar{s}_A^\top, 1)\bar{c} \bmod p \right\rceil_{0,1/2}$

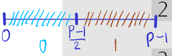


Construction 1

- *Key generation* $\mathsf{Gen}(1^n)$*:*
  1. *Sample matrix* $\bar{A} \leftarrow \mathbb{Z}_p^{n\times m}$ *for* $m, p = \mathrm{poly}(n)$
  2. *Sample secret key* $\bar{s}_A \leftarrow \mathbb{Z}_p^n$ *and error* $\bar{e} \leftarrow \mathrm{E}_\alpha^m$
  3. *Output* $\left(\mathrm{pk} := \begin{pmatrix} \bar{A} \\ \bar{t}_A^\top \end{pmatrix}, \mathrm{sk} := \bar{s}_A\right)$*, where* $\bar{t}_A^\top := \bar{s}_A^\top \bar{A} + \bar{e}^\top \bmod p$

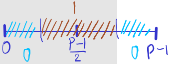


- *Encryption* $\mathsf{Enc}(\mathrm{pk}, b)$*:*
  1. *Sample random coin* $\bar{s}_B \leftarrow \{0,1\}^m$
  2. *Encode message* $\tilde{b} := b \cdot \lfloor p/2 \rceil$
  3. *Output* $\bar{c} := \mathrm{pk}\bar{s}_B + \begin{pmatrix} 0^n \\ \tilde{b} \end{pmatrix} \bmod p$



$\begin{pmatrix} \bar{A}\bar{s}_B \\ \bar{t}_A^\top \bar{s}_B \end{pmatrix}$

$-\bar{s}_A^\top \bar{A}\bar{s}_B + \bar{t}_A^\top \bar{s}_B$



- *Decryption* $\mathsf{Dec}(\mathrm{sk}, \bar{c})$*: output* $\left\lfloor (-\bar{s}_A^\top, 1)\bar{c} \bmod p \right\rceil_{0,1/2}$
- *Correctness of decryption: similar argument to key exchange*

Regev's Encryption is Quantum Secret...

Theorem 3 (LWE → Quantum CPA-PKE)

Regev PKE is quantum CPA-secret under DLWE assumption.

Proof sketch. Hybrid argument with two steps.

Step 1: Real world H_0

$\downarrow$

Random world H_1

Theorem 3 (LWE $\rightarrow$ Quantum CPA-PKE)

Regev PKE is quantum CPA-secret under DLWE assumption.

Proof sketch. Hybrid argument with two steps.

Step 1: Real world H_0 : $pk = \begin{pmatrix} \bar{A} \\ \bar{t}_A \end{pmatrix}$, where $\bar{t}_A := \bar{s}_A^T \bar{A} + \bar{e}$

$\downarrow$

Random world H_1: $pk = \begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}$, where $\bar{r} \leftarrow \mathbb{Z}_p^m$

Theorem 3 (LWE → Quantum CPA-PKE)

Regev PKE is quantum CPA-secret under DLWE assumption.

Proof sketch. Hybrid argument with two steps.

Step 1: Real world H_0 : $pk = \begin{pmatrix} \bar{A} \\ \bar{t}_A \end{pmatrix}$, where $\bar{t}_A := \bar{s}_A^T \bar{A} + \bar{e}$

$\downarrow$

Random world H_1: $pk = \begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}$, where $\bar{r} \leftarrow \mathbb{Z}_p^m$

Claim1: H_0 is quantum indistinguishable from H_1 assuming DLWE.

Regev's Encryption is Quantum Secret...

Theorem 3 (LWE → Quantum CPA-PKE)

Regev PKE is quantum CPA-secret under DLWE assumption.

Proof sketch. Hybrid argument with two steps.

Step 1: Real world H_0: $pk = \begin{pmatrix} \bar{A} \\ \bar{t}_A \end{pmatrix}$, where $\bar{t}_A := \bar{s}_A^T \bar{A} + \bar{e}$

$\downarrow$

Random world H_1: $pk = \begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}$, where $\bar{r} \leftarrow \mathbb{Z}_p^m$

Claim 1: H_0 is quantum indistinguishable from H_1 assuming DLWE.

Proof: $\exists D'$ distinguisher for DLWE $\Leftarrow \exists D$ distinguisher for H_0 and H_1

Regev's Encryption is Quantum Secret...

Theorem 3 (LWE → Quantum CPA-PKE)

Regev PKE is quantum CPA-secret under DLWE assumption.

Proof sketch. Hybrid argument with two steps.

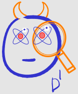
Step 1: Real world H_0: $pk = \begin{pmatrix} \bar{A} \\ \bar{t}_A \end{pmatrix}$, where $\bar{t}_A := \bar{s}_A^T \bar{A} + \bar{e}$

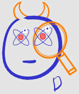
$\downarrow$

Random world H_1: $pk = \begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}$, where $\bar{r} \leftarrow \mathbb{Z}_p^m$

Claim 1: H_0 is quantum indistinguishable from H_1 assuming DLWE.

Proof: $\exists D'$ distinguisher for DLWE $\Leftarrow \exists D$ distinguisher for H_0 and H_1





Regev's Encryption is Quantum Secret...

Theorem 3 (LWE → Quantum CPA-PKE)

Regev PKE is quantum CPA-secret under DLWE assumption.

Proof sketch. Hybrid argument with two steps.

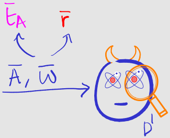
Step 1: Real world H_0: $pk = \begin{pmatrix} \bar{A} \\ \bar{t}_A \end{pmatrix}$, where $\bar{t}_A := \bar{s}_A^T \bar{A} + \bar{e}$

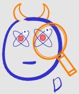
$\downarrow$

Random world H_1: $pk = \begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}$, where $\bar{r} \leftarrow \mathbb{Z}_p^m$

Claim1: H_0 is quantum indistinguishable from H_1 assuming DLWE.

Proof: $\exists D'$ distinguisher for DLWE $\Leftarrow \exists D$ distinguisher for H_0 and H_1





Regev's Encryption is Quantum Secret...

Theorem 3 (LWE → Quantum CPA-PKE)

Regev PKE is quantum CPA-secret under DLWE assumption.

Proof sketch. Hybrid argument with two steps.

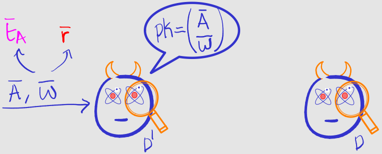
Step 1: Real world H_0 : $pk = \begin{pmatrix} \bar{A} \\ \bar{E}_A \end{pmatrix}$, where $\bar{E}_A := \bar{s}_A^T \bar{A} + \bar{e}$

$\downarrow$

Random world H_1: $pk = \begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}$, where $\bar{r} \leftarrow \mathbb{Z}_p^m$

Claim1: H_0 is quantum indistinguishable from H_1 assuming DLWE.

Proof: $\exists D'$ distinguisher for DLWE $\Leftarrow \exists D$ distinguisher for H_0 and H_1



Regev's Encryption is Quantum Secret...

Theorem 3 (LWE $\rightarrow$ Quantum CPA-PKE)

Regev PKE is quantum CPA-secret under DLWE assumption.

Proof sketch. Hybrid argument with two steps.

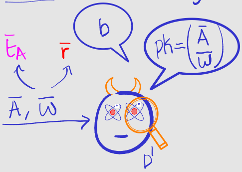
Step 1: Real world H_0: $pk = \begin{pmatrix} \bar{A} \\ \bar{E}_A \end{pmatrix}$, where $\bar{E}_A := \bar{s}_A^T \bar{A} + \bar{e}$

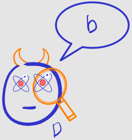
$\downarrow$

Random world H_1: $pk = \begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}$, where $\bar{r} \leftarrow \mathbb{Z}_p^m$

Claim 1: H_0 is quantum indistinguishable from H_1 assuming DLWE.

Proof: $\exists D'$ distinguisher for DLWE $\Leftarrow \exists D$ distinguisher for H_0 and H_1





Regev's Encryption is Quantum Secret...

Theorem 3 (LWE → Quantum CPA-PKE)

Regev PKE is quantum CPA-secret under DLWE assumption.

Proof sketch. Hybrid argument with two steps.

Step 1: Real world H_0: $pk = \begin{pmatrix} \bar{A} \\ \bar{t}_A \end{pmatrix}$, where $\bar{t}_A := \bar{s}_A^T \bar{A} + \bar{e}$

$\downarrow$

Random world H_1: $pk = \begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}$, where $\bar{r} \leftarrow \mathbb{Z}_p^m$

Regev's Encryption is Quantum Secret...

Theorem 3 (LWE → Quantum CPA-PKE)

Regev PKE is quantum CPA-secret under DLWE assumption.

Proof sketch. Hybrid argument with two steps.

Step 1: Real world H_0: $pk = \begin{pmatrix} \bar{A} \\ \bar{t}_A \end{pmatrix}$, where $\bar{t}_A := \bar{s}_A^T \bar{A} + \bar{e}$

$\downarrow$

Random world H_1: $pk = \begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}$, where $\bar{r} \leftarrow \mathbb{Z}_p^m$

Step2

Claim 2: PKE in H_1 is statistically secure.

Theorem 3 (LWE → Quantum CPA-PKE)

Regev PKE is quantum CPA-secret under DLWE assumption.

Proof sketch. Hybrid argument with two steps.

Step 1: Real world H_0: $pk = \begin{pmatrix} \bar{A} \\ \bar{t}_A \end{pmatrix}$, where $\bar{t}_A := \bar{s}_A^T \bar{A} + \bar{e}$

$\downarrow$

Random world H_1: $pk = \begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}$, where $\bar{r} \leftarrow \mathbb{Z}_p^m$

Step2

Claim 2: PKE in H_1 is statistically secure.

Proof: $c = \underbrace{\begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}}_{n+1 \times m} \bar{s}_B^{\,m} + \begin{pmatrix} 0^n \\ \tilde{b} \end{pmatrix}$

Theorem 3 (LWE → Quantum CPA-PKE)

Regev PKE is quantum CPA-secret under DLWE assumption.

Proof sketch. Hybrid argument with two steps.

Step 1: Real world H_0: $pk = \begin{pmatrix} \bar{A} \\ \bar{t}_A \end{pmatrix}$, where $\bar{t}_A := \bar{s}_A^T \bar{A} + \bar{e}$

$\downarrow$

Random world H_1: $pk = \begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}$, where $\bar{r} \leftarrow \mathbb{Z}_p^m$

Step 2

Claim 2: PKE in H_1 is statistically secure.

Proof:

$$c = \underbrace{\begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}}_{n+1 \times m} \overbrace{\bar{s}_B}^{m} + \begin{pmatrix} \bar{0}^n \\ \tilde{b} \end{pmatrix} \Rightarrow \begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix} \bar{s}_B \text{ loses information about } \bar{s}_B$$

Theorem 3 (LWE → Quantum CPA-PKE)

Regev PKE is quantum CPA-secret under DLWE assumption.

Proof sketch. Hybrid argument with two steps.

Step 1: Real world H_0: $pk = \begin{pmatrix} \bar{A} \\ \bar{t}_A \end{pmatrix}$, where $\bar{t}_A := \bar{s}_A^T \bar{A} + \bar{e}$

$\downarrow$

Random world H_1: $pk = \begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}$, where $\bar{r} \leftarrow \mathbb{Z}_p^m$

Step2

Claim 2: PKE in H_1 is statistically secure.

Proof: $c = \underbrace{\begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}}_{n+1 \times m} \overset{m}{\bar{s}_B} + \begin{pmatrix} 0^n \\ \tilde{b} \end{pmatrix} \Rightarrow \begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix} \bar{s}_B$ loses information about $\bar{s}_B$

Matrix "Leftover hash lemma": For $\begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix} \leftarrow \mathbb{Z}_p^{(n+1) \times m}$, $\left(\begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}, \begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix} \bar{s}_B \right) \approx_s \left(\begin{pmatrix} \bar{A} \\ \bar{r} \end{pmatrix}, \bar{r}' \leftarrow \mathbb{Z}_q^{n+1} \right)$ ∎

Regev's Encryption is Quantum Secret...

Exercise 2

Show that the following variants of Regev's scheme are also quantum secret assuming DLWE:

1. *Gaussian secret-keys: same as in Construction 1 except sample the secret key as* $\bar{\boldsymbol{s}}_{\boldsymbol{A}} \leftarrow \mathrm{E}_{\alpha}^{\boldsymbol{n}}$
2. *Gaussian random coins: same as in Construction 1 except sample the random coin as* $\bar{\boldsymbol{s}}_{\boldsymbol{B}} \leftarrow \mathrm{E}_{\alpha}^{\boldsymbol{m}}$

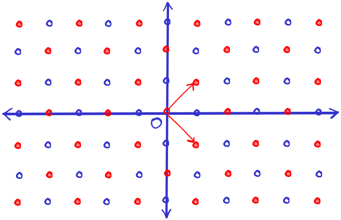
Plan for This Lecture...

1 Motivation: Quantum Adversaries

2 Learning with Errors (LWE)

3 Cryptography from LWE

4 LWE and Lattices



What has LWE to Do with Lattices?...

Defintion 1 (Lattice)

A n-dimensional lattice $\mathbb{L}$ is a discrete, additive subgroup of $\mathbb{R}^n$.

What has LWE to Do with Lattices?...

Defintion 1 (Lattice)

"points sufficienty far apart"

A n-dimensional lattice $\mathbb{L}$ is a discrete, additive subgroup of $\mathbb{R}^n$.

subset that is also group

What has LWE to Do with Lattices?...

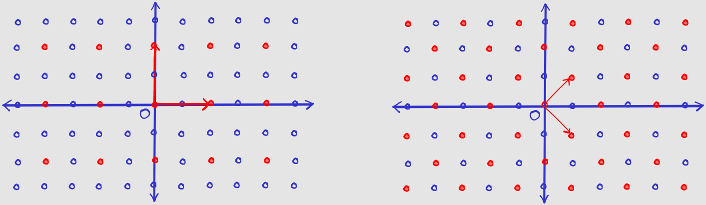
Defintion 1 (Lattice)

"points sufficiently far apart"

A n-dimensional lattice $\mathbb{L}$ is a discrete, additive subgroup of $\mathbb{R}^n$.

subset that is also group

Example 4 (2D scaled integer and 2D checkerboard lattice)



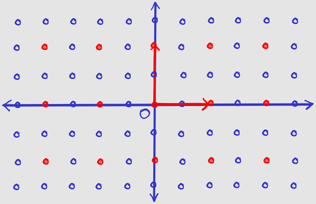
What has LWE to Do with Lattices?

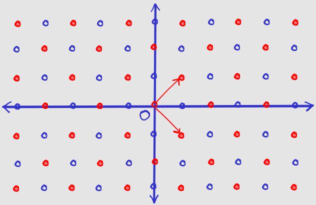
Defintion 1 (Lattice) "points sufficienty far apart"

A n-dimensional lattice $\mathbb{L}$ is a discrete, additive subgroup of $\mathbb{R}^n$.

subset that is also group

Example 4 (2D scaled integer and 2D checkerboard lattice)





- Represented using a basis $\bar{B} = (\bar{b}_1, \cdots, \bar{b}_n) \in \mathbb{R}^{n\times n}$ as its integer linear combination:

$$\mathbb{L}(\bar{B}) := \left\{ \bar{v} := \sum_{i\in[n]} a_i \bar{b}_i \text{ for } (a_1, \ldots, a_n) \in \mathbb{Z}^n \right\}$$

What has LWE to Do with Lattices?...

- Some *worst-case* hard problems on lattices:
  1. Shortest vector problem (SVP)
     - Input: lattice $\mathbb{L}$ via basis $\bar{B}$
     - Solution: shortest (in 2-norm) non-zero vector in $\mathbb{L}$

What has LWE to Do with Lattices?...

- Some *worst-case* hard problems on lattices:
  1. Shortest vector problem (SVP)
     - Input: lattice $\mathbb{L}$ via basis $\bar{B}$
     - Solution: shortest (in 2-norm) non-zero vector in $\mathbb{L}$
  2. GapSVP$_\gamma$: decision version of SVP
     - Input: lattice $\mathbb{L}$ via basis $\bar{B}$, and $d \in \mathbb{R}$
     - Decide whether the shortest vector has length $\geq \gamma d$ or $< d$
     - NP-hard for constant γ, but not for $\gamma = \mathsf{poly}(n)$

What has LWE to Do with Lattices?...

- Some *worst-case* hard problems on lattices:
  1. Shortest vector problem (SVP)
     - Input: lattice $\mathbb{L}$ via basis $\bar{B}$
     - Solution: shortest (in 2-norm) non-zero vector in $\mathbb{L}$
  2. GapSVP$_\gamma$: decision version of SVP
     - Input: lattice $\mathbb{L}$ via basis $\bar{B}$, and $d \in \mathbb{R}$
     - Decide whether the shortest vector has length $\geq \gamma d$ or $< d$
     - NP-hard for constant γ, but not for $\gamma = \mathsf{poly}(n)$

Theorem 5 (Worst-case to average case reduction)

Solving $(n, m, p, \mathrm{E}_\alpha)$-LWE, for $\alpha p \geq \sqrt{n}$, in the average case is at least as hard as deciding GapSVP$_{\tilde{O}(n^2)}$ for any n-dimensional lattice $\mathbb{L}$

What has LWE to Do with Lattices?...

- Some *worst-case* hard problems on lattices:
  1. Shortest vector problem (SVP)
     - Input: lattice $\mathbb{L}$ via basis $\bar{B}$
     - Solution: shortest (in 2-norm) non-zero vector in $\mathbb{L}$
  2. GapSVP$_\gamma$: decision version of SVP
     - Input: lattice $\mathbb{L}$ via basis $\bar{B}$, and $d \in \mathbb{R}$
     - Decide whether the shortest vector has length $\geq \gamma d$ or $< d$
     - NP-hard for constant γ, but not for $\gamma = \mathrm{poly}(n)$

Theorem 5 (Worst-case to average case reduction)

Solving $(n, m, p, \mathrm{E}_\alpha)$-LWE, for $\alpha p \geq \sqrt{n}$, in the average case is at least as hard as deciding GapSVP$_{\tilde{O}(n^2)}$ for any n-dimensional lattice $\mathbb{L}$

- Compare with factoring
  - Only weakly one-way and most instances are easy
  - Worst-case to average case reduction not known

To Recap Today's Lecture

- Discussed motivation for post-quantum cryptography

To Recap Today's Lecture

- Discussed motivation for post-quantum cryptography
- Introduced a new hardness assumption: LWE
  - Saw equivalence between its decision and search variants
  - Constructed key-exchange protocol from DLWE

To Recap Today's Lecture

- Discussed motivation for post-quantum cryptography
- Introduced a new hardness assumption: LWE
  - Saw equivalence between its decision and search variants
  - Constructed key-exchange protocol from DLWE
- LWE has enough "structure" to support more advanced cryptographic primitives:
  1. Fully-homomorphic encryption (FHE): coming up, Lecture 19(?)
  2. Identity-based encryption: PKE where the public keys are arbitrary strings
  3. Incrementally-verifiable computation ...

To Recap Today's Lecture

- Discussed motivation for post-quantum cryptography
- Introduced a new hardness assumption: LWE
  - Saw equivalence between its decision and search variants
  - Constructed key-exchange protocol from DLWE
- LWE has enough "structure" to support more advanced cryptographic primitives:
  1. Fully-homomorphic encryption (FHE): coming up, Lecture 19(?)
  2. Identity-based encryption: PKE where the public keys are arbitrary strings
  3. Incrementally-verifiable computation ...
- Related computational problem: learning *parity* with noise
  - "Modulus 2 version" of LWE
  - Open: PKE from LPN

Next Lecture

- So far in Module II: secrecy in the public-key setting

Next Lecture

- So far in Module II: secrecy in the public-key setting

- Next lecture: integrity + authentication in *public-key* setting
- New cryptographic primitive: *digital signatures*
  - Two construction, both quantum secure
    - Lamport's one-time signature $\leftarrow$ OWF
    - Theoretic construction of stateless signature
  - New proof technique: plug and pray!

References

1. [KL14, §14.3] for details of this chapter
2. For a formal introduction to quantum computing, use [NC10]; a quick introduction can be found in [AB09, Chapter 10] (including Grover's and Shor's algorithms)
3. For a formal introduction to lattice-based cryptography, refer to Peikert's survey [Pei16] or lecture notes of Vaikuntanathan's CS294 course.
4. The LWE-based encryption in Construction 1 is from [Reg05], but the presentation is from [Pei16, §5.2.1]
5. The worst-case to average-case reduction for LWE in the form stated in Theorem 5 is due to a series of works: [Reg05, Pei09, LM09]

Sanjeev Arora and Boaz Barak.
Computational Complexity - A Modern Approach.
Cambridge University Press, 2009.

Jonathan Katz and Yehuda Lindell.
Introduction to Modern Cryptography (3rd ed.).
Chapman and Hall/CRC, 2014.

Vadim Lyubashevsky and Daniele Micciancio.
On bounded distance decoding, unique shortest vectors, and the minimum distance problem.
In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 577–594. Springer, Heidelberg, August 2009.

Michael A. Nielsen and Isaac L. Chuang.
Quantum Computation and Quantum Information: 10th Anniversary Edition.
Cambridge University Press, 2010.

Chris Peikert.
Public-key cryptosystems from the worst-case shortest vector problem: extended abstract.
In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 333–342. ACM Press, May / June 2009.

Chris Peikert.
A decade of lattice cryptography.
Found. Trends Theor. Comput. Sci., 10(4):283–424, 2016.

Oded Regev.
On lattices, learning with errors, random linear codes, and cryptography.
In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.