https://hackmd.io/@haopin/BJ-qErR\_a

先 high level 了解一下 lattice 是什麼以及會用到的難問題

https://www.youtube.com/@chalktalkmath

Post-quantum cryptography: Security after Shor’s algorithm

Lattice-based cryptography: The tricky math of dots

Learning with errors: Encrypting with unsolvable equations

A Decade of Lattice Cryptography

https://web.eecs.umich.edu/~cpeikert/pubs/lattice-survey.pdf

先看 1-4 章

----------------------------------------------------------------------------------

1/18

youtube 影片1

開頭講NIST舉辦contest去找後量子密碼的加密標準 公開金鑰加密基於數學上的難問題

未來Shor’s algorithm 搭配足夠穩定 足夠多qubits的量子電腦時 對目前的公開金鑰加密會有問題

比較 1.Post-Quantum Cryptography (PQC):對抗量子電腦的加密標準

2.Quantum Cryptography:利用量子力學在傳統電腦上加密

MIST開發新標準的目標是要滿足原本公開金鑰安全和後量子安全的

後量子密碼學主要方向:

1.Lattice-based Cryptography:

基於格的加密算法使用數學上的格結構，其中格是向量空間的離散子集。

這些算法的安全性基於某些格問題的難解性，

例如SVP（Shortest Vector Problem）和LWE（Learning With Errors）。

Lattice-based 加密被視為一個強大且有潛力對抗量子計算攻擊的方向。

2.Code-based Cryptography:

基於代碼的加密算法使用線性代數和錯誤更正碼的概念。

這些算法基於破解一些特定編碼問題的難解性，例如找到矩陣的最小距離。

McEliece 加密是基於代碼的一個例子

3.Hash-based Cryptography:

基於哈希的加密算法主要依賴於哈希函數的性質。

Merkle-Damgård構造是基於哈希的加密的一個例子，其中安全性基於哈希函數的阻撓性。

Winternitz One-Time Signature Scheme是基於哈希的一個簽名方案。

4.Multivariate Cryptography:

多變數加密使用多變數多項式系統，這些多變數多項式系統的求解被假設為困難的。

其中的一個例子是Multivariate Quadratic Equations (MQ) 簽名方案

5.Supersingular Isogeny Cryptography:

超奇異同源加密是一種基於橢圓曲線同源理論的新型加密方法。

這種方法的安全性基於求解超奇異同源問題的困難性。

SIDH（Supersingular Isogeny Diffie-Hellman）是超奇異同源加密的一個例子。

----------------------------------------------------------------------------------

youtube影片2

lattice的定義

格（lattice）是指由基向量構成的離散的、無窮網格的集合。

二維格（lattice）可以由兩個線性獨立的基向量構成。如果這兩個基向量分別是 v1 v2

這個格中的每個點都可以表示為c1v1+c2v2 , c1 c2為整數

格密碼學中通常是處理高維格 基向量的選擇和格的性質影響數學問題的難解性

SVP (Shortest Vector Problem) 最短向量問題

LWE(Learning With Errors) 學習與錯誤問題

這些問題的困難性是格密碼學的基礎

c1v1+c2v2 => 可想成從原點移動c1次v1 移動c2次v2 移動的次序不影響結果

也就是說 (-1)\*v1+2\*v2+1\*v1 等價於 2\*v2 同一個vector的係數可以合併

basis vector 改變會導致整個lattice points跟著移動

不同的basis vector可能產生相同的lattice 主要重點會在生成子空間(span)

如果兩組基向量的span相同就會產生相同的格 =>用單組向量的線性獨立判斷

ex: (7,3) (2,1) 找有沒有a(7,3)+b(2,1)=(0,0)的非0解 =>沒有

代表(7,3) (2,1)組成的格和(0,1) (1,0)組成的格相同 (整數格)

在二維只有單個點(有0向量) 和 一條線(v1平行v2) 兩種情況會有問題 其他都會是整數格((1,0)和(0,1)組成的)

shortest vector Problem:找離原點最近的線性組合 在二維是簡單的 因為只要線性獨立且兩向量不平向 任意的基礎向量都可以是

離原點最近的結果 二元一次找出線性組合就可以了。

但高維

closet vector problem:找離指定的點最近的點 minimize ∥v−t∥ (歐氏距離)

GGH encryption scheme (Goldreich Goldwasser Halevi) =>

public key:bad basis (難計算) 用來產生lattice (用2維舉例的話兩個向量可能更接近平行導致更難計算)

private key:good basis (好計算) 用來解密

明文會先映射在一個點上面 加密時將點稍微移動變成密文 使最接近的點仍然為密文的那個點 (SVP算出的答案會是明文)

bad basis解密非常困難 good basis容易解SVP

----------------------------------------------------------------------------------

1/19

youtube 影片3

learning without errors => 可以想成有一些整數為private key

這些private key 搭配一些不同係數產生出的一些等式作為public key

ex:

private key: x=1, y=2, z=3, w=4

public key : {

2x+y+2z+w=14

x+y+z+w=10

...........

...........

...........

}

安全性很低=>因為多元一次很好解(高斯消去)

learning with errors(LWE)

在等式的右邊加一些errors => 讓攻擊者無法從多元一次解出準確的private key的值

error通常隨機產生(值偏小) 不包含在private key裡 public也看不到

搭配mod的概念 可以想成出來的值會是一圈的

ex: 加密0或是1的bit

加密方法:挑幾個LWE的等式相加 產生出的結果是actual solution + encoded bit

actual solution可以透過private key去算等號左邊得到

加密0的時候把用公鑰挑的幾個等式相加以後的原始結果放在右邊

加密1的時候把用公鑰挑的幾個等式相加以後的結果加上模的一半放在右邊

解密時用private key算出的結果偏向原始結果或是加過模的一半的結果來判斷密文是0或是1

lattice結合LWE =>利用error對密文對應的點做偏移(加密時引入error)

=> shortest vector problem(SVP)的困難性提高安全

----------------------------------------------------------------------------------

A Decade of Lattice Cryptography 書

Abstract :

Lattice-based cryptography : 以point lattices in R^n 的 hard problems為基礎for secure

近十年的重點在short integer solution (SIS)和 learning with errors (LWE) problems

CH1

1.Conjectured security against quantum attacks:

Shor algorithm讓我們在未來large-scale quantum computers are available的情況下RSA和Diffie

Hellman 都變得不安全。

對格密碼學來說no efficient quantum algorithms are known for problems in格密碼的難問題

(目前量子電腦無法speedup格密碼的難問題)

2.Algorithmic simplicity, efficiency, and parallelism:

格密碼的演算法通常簡單 有效率 高度平行化 很多linear operations on vectors and matrices modulo

relatively small integers

3.Strong security guarantees from worst-case hardness:

//這裡的worst case和演算法的worst case不同 指的是對攻擊者最有利 需要花最少時間的情況下

對攻擊者最有利的情況下密碼仍然很安全才被認為是一個安全的密碼

4.Constructions of versatile and powerful cryptographic objects:

(1) Fully Homomorphic Encryption (FHE) : 允許一個不被信任的工作者對加密的數據進行計算，且仍然部會

獲得和數據相關的任何信息。

(2)Attribute-based Encryption (ABE) : 允許指定符合某些屬性或條件的用戶才能解密數據

(3)Code Obfuscation : 代碼混淆

這些都是格密碼學的優勢

1.1 scope and organization

研究重點在於short integer solution (SIS) and learning with errors (LWE) problems

章節重點 第二章:基礎數學和密碼學背景

第三章:high level的概述和重要性

第四章:現代基礎，正式定義SIS LWE問題，worst cse同樣很難解決的定理，更緊湊和高校的基於環

的類比 ring-SIS ring-LWE。

第五章:格密碼學的構造，加密和數位簽章 概念和技術

第六章:更高級的密碼學構造，重點在完全同態加密

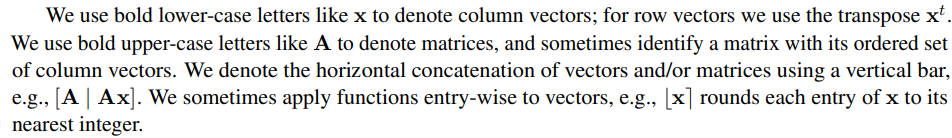
第七章:格密碼學領域中的重要開放問題

----------------------------------------------------------------------------------

CH2



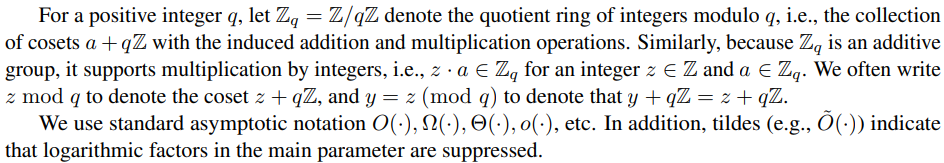
取高斯=> 小於x的最大整數 後面那個是最接近的整數 (用+0.5判斷他是否>=0.5)



x是列向量(直的column )， x^t是行向量(橫的 row) ，A表示矩陣

[A | Ax] 表示矩陣 A 和 Ax 的橫向串聯

Round x表示把向量x的每個元素做四捨五入



//

群G(Group)

要滿足GxG->G 封閉性 結合律

存在單位元素=> axe=exa=a ，e就是群的單位元素

反元素 a x a^-1 =a^-1 x a=e ， a^-1為a的逆元素

//

Zq = Z/qZ 表示整數模 q 的商環，也就是1,2,…,q-1的集合(這邊的Z/qZ出來的結果是餘數)

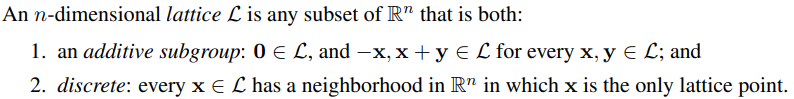
z mod q 代表 z + qZ 的餘類 同餘=>y=z(modq)

Zq是一個加法群

使用標準的漸進符號 O(·)、Ω(·)、Θ(·)、o(·) 等，這些符號用來描述算法的時間複雜度或函數的增長率。

在 O˜(·) 中，波浪號表示對主要參數的對數因子進行了省略，即將對數因子忽略不計。

Lattice:



n-dimensional lattice 包含了0 然後他是加法群=>群的元素相加或加負號也會在群內。

Discrete講的是有一個點和點之間最短距離的限制(可想成是一個球面 球面上可能有多個點)，不會過於密集

一些格的例子

1. integer lattice Z^n
2. scaled lattice cL for any real number c and lattice L
3. “checkerboard” lattice(xi限制為偶數) 類似棋盤上交替方塊的結構



Lattice L的minimum distance由shortest nonzero lattice vector決定(L代表lattice

 (意思的向量長度

e ith successive minimum λi(L) is the smallest r such that L has i linearly independent vectors of norm at most r.

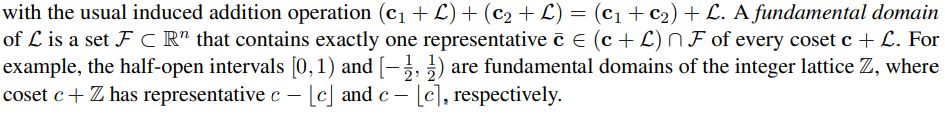
有λ0(L), λ1(L),…, λi-1(L)向量長度越來越大且互相線性獨立，這些向量長度都不超過r。

Lattice的概念加上乘法群的概念

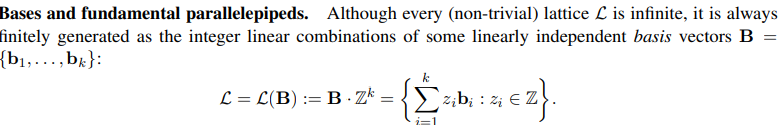
Lattice是加法群可以做出quotient group R^n/L (像是點除lattice的餘數)



(c1 + L) + (c2 + L) = (c1 + c2) + L 可以想成把空間切成多個(lattice和他的右上角的部分) 所有空間中的點會對應到區域內的部分 =>這個概念類似於fundamental domains



Fundamental domain(F):在R^n/L下所有可能性所構成的集合(不能重複)，有多種可能的表達方式。 c bar叫做F中的代表元素，空間內的每個點對應到一個代表元素。



Finitely generated(有限生成)=>點有無限個但都還是透過基礎向量的線性組合產生，而非毫無規則的密集。 L(B)是以B為basis生成的格子

k是basis的rank 大部分討論full-rank lattices (k=n)

unimodular matrix  (determinant= +/-1)

B．U也是basis of L(B) 因為determinant=1不會做縮放 只是把座標作旋轉而已，原本格的性質不會變(距離、密集度等)

旋轉了座標會改變 ，但格密碼學中真正的座標其實沒那麼重要。重點會放在格子的基本性質 ex: 基底的長度、基底之間的角度、格子的密度等等。可以更專注在基底和基底之間的線性組合。



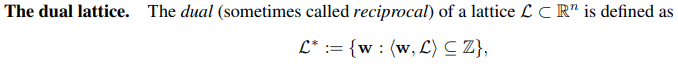
對B為basis的基本域

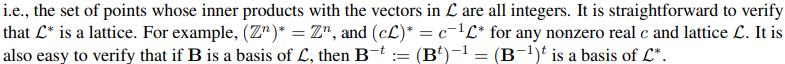
origin-centered fundamental parallelepiped : P(B) 每個維度的-1/2到1/2

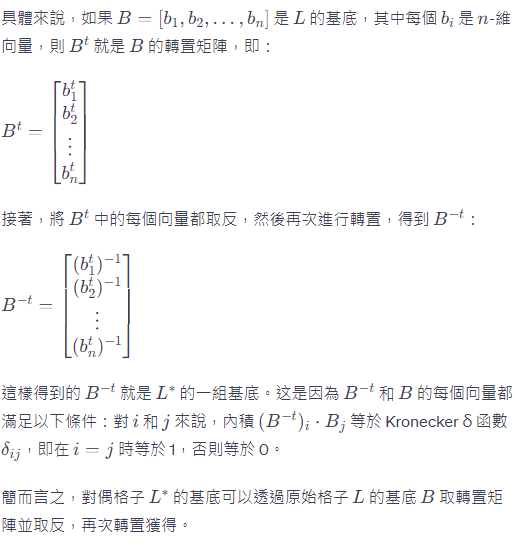
c+L可以表示為

是在把c轉換成基底B的座標表示，用投影的方式找到離c最近的點。

在拿c減掉上面投影的結果就是基底B下的代表元素





dual (reciprocal) of a lattice L :一個集合符合和L內積的結果都是整數，有對應的basis會等於B^-t 

2.2.2 Computational Problems



給一些格子，找他所有basis的可能中的最短的非零向量，np-hard



一樣給了一些格子 找一個長度小於γ(n)\* λ1(L)的向量

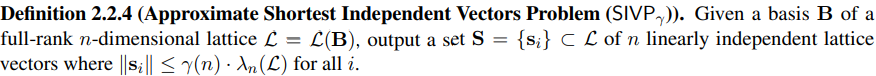
(γ是和格子的維度n相關的一個函數，即γ = γ(n)。)

λ1(L)是用一些方法推測出來的，γ(n)把問題難度降低一點點，擴大可接受的範圍。 但γ(n)可能很小，難度還是有可能很高

單純的SVP是安全的 但SVPγ的安全性還沒被證明



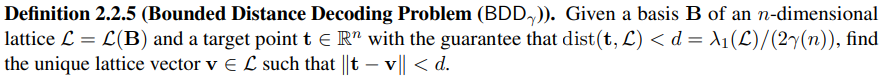
判斷basis B生成的格子L = L(B)的最短向量λ1(L)是否滿足範圍λ1(L) ≤ 1 或 λ1(L) > γ(n)。 通常應用在安全性證明



SVPγ的延伸版，變成要找一組符合長度限制且線性獨立的最短向量。等同找一組符合長度限制的basis vector

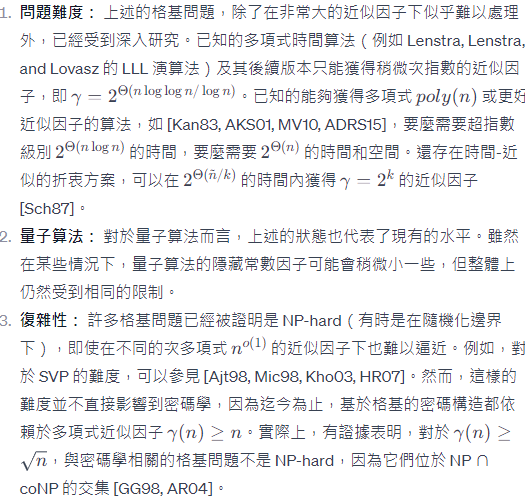
//CVPγ (closest vector problem) 給一個任意點找最接近的格子點，沒有限制範圍，可能有多個最接近的格子點=>可能有多組解

BDDγ (Bounded Distance Decoding Problem) 有界距離解碼問題，會要求目標點很接近格子點，以一定的有界距離約束問題，密碼學更多會用BDDγ



t是給的點(target point)要小於d = λ1(L)/(2γ(n)) 要找的格子點符合||t-v|| < d是唯一的。

Algorithms and complexity



2.3 (Discrete) Gaussians and Subgaussians

先跳過p.11 12還沒讀完

2.4 Cryptographic Background

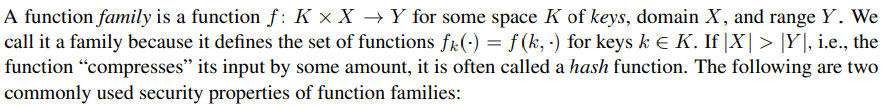
在複雜度理論密碼學中，安全參數λ(security parameter)調節所有計算的時間複雜度和攻擊者的成功機率(advantage優勢)

通常要求計算時間polynomial  in the security parameter，攻擊者的優勢是可忽略的

**O(1)** 和 **−ω(1)** 都是常數因子一個正一個負表示λ越大時間越長 破解機會越低。

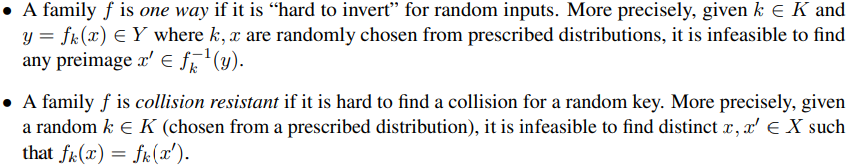
//複雜度O上界、 ω下界 、Θ確認上下界

2.4.1



由K的keys(k)決定從X轉換到Y的方法，每個k都有相應的函數

hash function=> |X|>|Y|有壓縮input的效果



one way(單向性): 給y=fk(x)，很難找到x

collision resistance(抗碰撞性): 很難找到不同的x和x’符合fk(x)=fk(x’) ，不同輸入產生相同值的機率很低

2.4.2 public key encryption (asymmetric encryption)

分成三部分1.key generator: 給security parameter output public/secret key

2.encryption algorithm拿public key和message output ciphertext

3.decryption algorithm拿secret key和ciphertext output message or distinguished failure

Correct: decrypt(encrypt(M))=M

實驗: 只有1 、 0兩種可能給密文判斷明文 ， 猜對的機率要非常低

兩種安全性

semantic secure和actively secure

1. semantic secure: 只能從密文觀察

2. actively secure: 攻擊者可選任意明文得到密文

Chosen plaintext attack是公開金鑰基本的要求

2.4.3 richer forms of encryption

Homomorphic encryption=>可以對加密的數據進行有意義的計算，不須解密。

Identity-based encryption(IBE)身分驗證加密

任何字串都可以當公鑰 包含4個tuple of randomized algorithms

1. setup algorithm:給security parameter，output a master public key and secret key pairs。
2. key extraction:用master secret key和identity string，output a secret key for particular identity。
3. encryption algorithm:用master public key和identity string和a valid message，output ciphertext。
4. decryption algorithm:用identity secret key和ciphertext，output massage

主公鑰用來加密，主私鑰用來產生每個特定身分的私鑰。加密用主公鑰和身分信息，解密用自己的私鑰。

Correctness: 用自己的私鑰解密(用公鑰和身分信息加密(M))=M

IBE的semantic security:攻擊者有主公鑰且有演算法的訪問權限，可以任意選擇身分信息獲得私鑰，m0 m1用同一個身分識別加密時攻擊者要無法分辨。

Ch3

* 1. Ajtai’s Function and Ajtai-Dwork Encryption

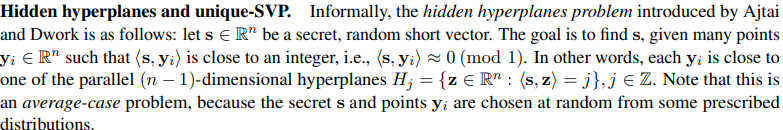
Ajtai首次提出格問題的worst case和average case格問題的歸納，第一個計算格上的難問題的安全性證明。引入了SIS(Short integer solution)problem和一些相關的one way function，也證明了他至少和approximating various lattice problems in the worst case一樣困難。SIS 和 Ajtai’s function很常被用在密碼學中。4.1節

Ajtai and Dwork提出了lattice based的public key encryption。所有的格加密都繼承這個系統的模板。證明了hidden hyperplanes problem(隱藏超平面問題)的average case至少和γ-unique shortest vector problem(uSVPγ) γ是多項式(poly(n))

他們也建構了一個公鑰加密系統，可以基於隱藏超平面問題的難度來證明語義安全性，因此也可以基於猜測最壞情況下的uSVPγ的難度

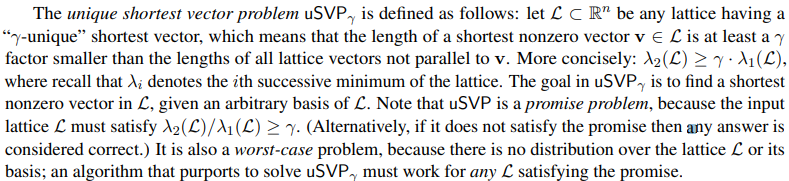
Ajtai and Dwork的缺點在於key size太大 公鑰O(n^4) 私鑰和密文O(n^2)

Ex: 加密128位的對稱密鑰 需要密文大小為數兆位。



隱藏超平面問題: 給定一些點yi，要找到一個短向量s，使yi和s構成的平面H的投影(內積)盡可能靠近整數=>可以想成讓yi盡可能靠近平面H，因為內積後為整數=>讓夾角盡量靠近0，cos夾角的值就趨近1，投影就趨近原本的整數值

公開的資訊是yi的座標和<s,yi>接近整數(可能不會給接近的那個整數值) 要求s



The unique shortest vector problem(uSVPγ):一樣是在找最短向量，但多了

λ₂(L) ≥ γ · λ₁(L)的限制來確保他最短。

HHP用在cryptosystem: 針對單個bit，如果加密0挑隨積點=>離平面遠=>結果不接近整數。如果加密1挑一個yi=>靠近平面=>結果接近整數。由此判斷是加密1或0

它的安全性證明是利用把這項加密方法轉換成相當於解HHP的問題 所以安全性至少和HHP一樣困難 除此之外還可以轉換成解決uSVPγ 的worst case所以難度也和這個相關。

* 1. NTRU 第一個用polynomial rings的加密結構

key比Ajtai-Dwork Encryption更緊湊，參數好的時候安全性也夠高(早期的key太緊湊導致安全性不夠)

NTRU沒有的安全性理論研究比較少。NTRU的variant已經被證明安全，難度和ring-LWE有關 section5.2.4

NTRU透過一個多項式環R= Z[X]/(f(X))做參數化 例如n為prime=> f(x) = x^n - 1

n為power of 2=>f(x) = x^n + 1 第一次範圍縮小)

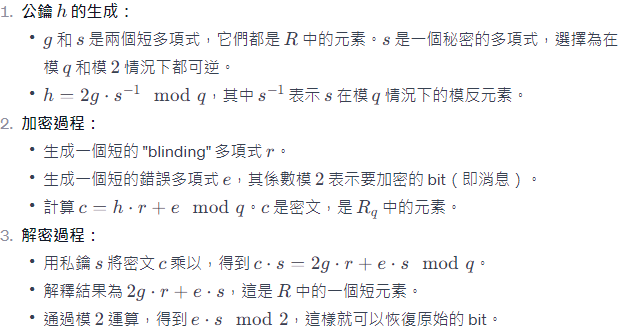
取一個足夠大的奇數q 使商環Rq=R/qR 第二次範圍縮小)

取兩個短多項式s g(整數係數小)，s是secret key，public key是h = 2g · s^−1 ∈ Rq

s對mod q和mod 2都可逆 =>s一定是奇數 (因為mod x這件事只有和x互值的數才有反元素)

encryption: ciphertext c = h · r + e ∈ Rq ，h是公鑰，r∈ R是short blinding factor，e是short error term

decryption:拿ciphertext\*secret key => c\*s=h\*r\*s+e\*s= 2g\*r+e\*s ∈ Rq



製造公鑰時的刻意用g和s^-1是為了隱藏s的資訊 然後用s^-1是為了之後解密能消掉s項 然後錯誤多項式e是為了拿來加密bit的 e\*s mod2的結果能夠表達1或0是因為s有反元素所以一定是奇數

* 1. Goldreich, Goldwasser, and Halevi Encryption and Signatures (GGH)

GGH的核心思想是公鑰是lattice的bad basis，私鑰是lattice的good basis

Bad basis=>consists of long and highly non-orthogonal lattice vectors

Good basis=>consists of relatively short lattice vectors

Bad basis可以從good basis用unimodular transformation轉換來

每個整數格子都有一個特別的basis: Hermite normal form =>他可以從任何基底有效計算出來，Hermite normal form是公共基的最好選擇。

原始GGH不安全，但概念重要

GGH encryption的概念前面有講過，秘密是lattice v ∈ L 加密時用small error e ∈ R^n ， 密文c = v + e ∈ R^n，因為error很小所以密文會非常接近v代表唯一解

bounded-distance decoding problem=>限制範圍的找最近點

GGH signature的概念相反，原訊息是公開的不在格子上的點，然後簽名是把他映射到最近的格子點上。公鑰提供bad basis驗證

簽章會有安全性問題=>會洩漏good basis的訊息

NTRU結合GGH容易受到安全性攻擊，後來增加了perturbation(擾動)的技術，透過增加密鑰和參數來減少簽名對密鑰的洩漏。

* 1. Micciancio’s Compact One-Way Function

Micciancio [Mic02]:Mic02對Ajtai的單向函數做修改適應R = Z[X]/(Xn − 1)多項式環，密鑰大小和運行時間從O(n^2)縮小到O(n)，修改後的函數有單向性但沒有抗碰撞性。

* 1. Regev’s Improvements to Ajtai-Dwork

Regev改善了Ajtai和Dwork，最特別的是引入Gaussian measures 高斯測度(probability distributions) and harmonic analysis調和分析

要破解密碼需要能夠對 n 維格進行 uSVPγ（其中 γ 是一個大的多項式，約為 n^7）的求解，uSVPγ 會隨著 γ 的減小變得更難解

Ch4

主要在講格密碼學的基礎工作 特別是SIS和LWE和他們在環上的類比，和離散高斯機率分布的技術

4.1 short integar solution(SIS)

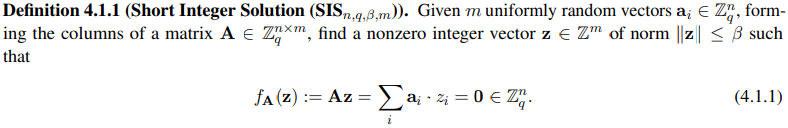
Ajtai首先提出的，有單向性和抗碰撞性

* + 1. definition

非正式的SIS=>給一些整數要找一個最小的線性組合=0

Ex: 2 3 7 5 =>2a+3b+7c+5d=0 => (a,b,c,d)=(1,1,0,-1)

正式定義



m個向量分別是ai構成矩陣A，要找線性組合z，使Az=0。 n是維度，q是mod數，β用來限制z的大小讓解足夠short，|z|<=β。 且q > β

observations:

1. 如果沒有*β*限制A的大小會太容易找到解，一定要有β < q不然z = (q, 0, . . . , 0)永遠都會是解 (因為會mod q)
2. 對矩陣A的任何解都可以透過附加零向量轉換成任何擴展[A|A0]，代表可以按需求忽略ai，因此SIS問題隨著m(給的向量數)增加變簡單。隨著n(維度)增加變困難。

3. β和m要夠大才能確定有解，m’是大於n log q的最小整數，β ≥ √ (m’)且m ≥ m’。假設沒有失去一般性，m=m’ (mod q下每個維度上有q種可能n維度=>q^n 但真實的每個向量不一定都是在同一個mod q區域 下的)

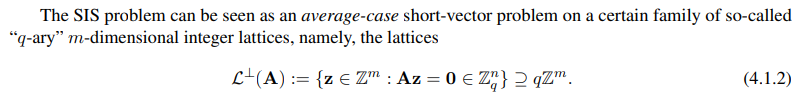
//鴿巢原理(pigeonhole argument):假設有q^n個鴿巢(範圍) 如果有q^n+1隻鴿子(向量)就一定會至少有一個巢有兩隻鴿子。所以如果m<q^n+1就一定存在兩個不同的向量x和x’使Ax=Ax’ 也就是( x - x’ )是A的一個非零解，且會小於β

=>

(這邊是在講要如何選向量數m比較好 然後特別提到了m bar(n log q往上取一個整數) 這個數量的m用鴿籠原理可以確保至少會有一個解滿足|z|<=β 和q > β 的範圍限制) m受到n q的限制讓問題一定有解且難度足夠。

4. SIS的難度是基於他的抗碰撞性

SIS引入到格問題上



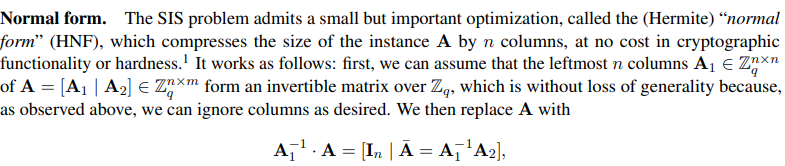
L ⊥(A)是所有可能z的集合，也就是符合Az=0的線性組合的所有可能，SIS就是在找離原點最近的格子點。 他的qZ m是指q的整數倍向量(aq,bq,cq)這種

inhomogeneous version of the SIS problem:把0改成一個隨機的u向量

變成是要找=u的線性組合

=u的解集合是=0的解集合的陪集，可以透過平移=0的解變成=u的解，所以這兩個版本的SIS難度應該等價。 可以想成是從shortest vector problem變成closest vector problem，

//矩陣行列 行是row 列是column



(Hermite) Normal form

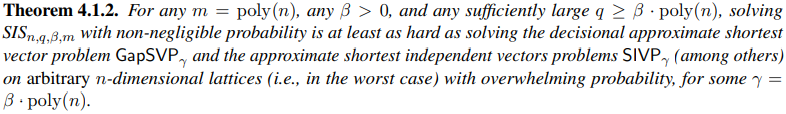
把矩陣A mxn 切成兩塊 A1 m\*m和A2 m\*(m-n)然後算出A1^-1。

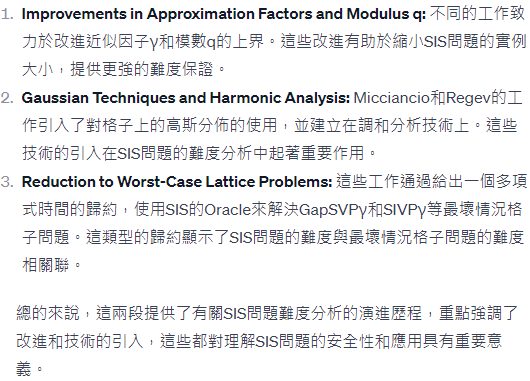
藉由A1^-1 \* A= A1^-1 \* [A1|A2]=[I m\*m|A1^-1 \* A2] 把需要考慮的向量數減少

求出解以後再把他還原，還原通常只需要做A1^-1 \* z’會得到原本的z。

//這種情況通常只是簡化計算，問題的難度理論上要不變。

* + 1. Hardness





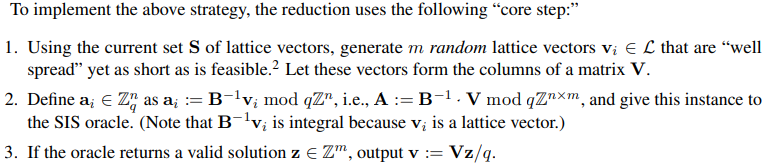
Reduction: 主要是在把worst case換成average case

Worst case可以想成是原本對長度沒限制的SIS，average case(SIVPγ)想要盡量靠近最小解。

原本有的是worst case的SIS Oracle，首先要找到所有線性獨立的格子向量集合S，每輪嘗試把最大向量長限制為上一輪的一半|S’|<=|S|/2直到無法執行。

無法執行是指長度已經夠小 長度小於SIVPγ的γ

Core step:每輪做的事

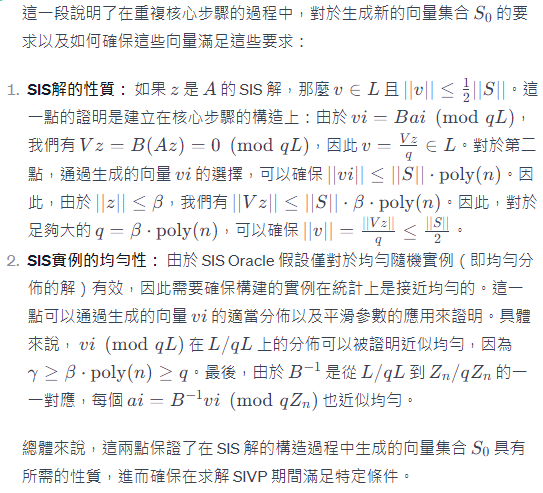


所以他每輪是隨機選(均勻分布的選)一些符合長度限制的向量然後丟到SIS Oracle看看有沒有回應 如果有就把S替換掉然後繼續嘗試下一輪

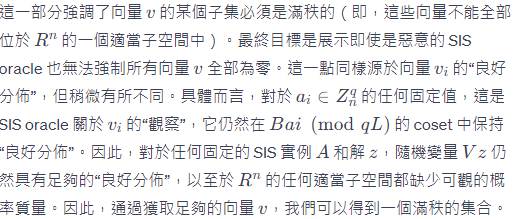
SIS Oracle是不一定存在的神奇演算法

p.23的那些黑點項目還沒讀

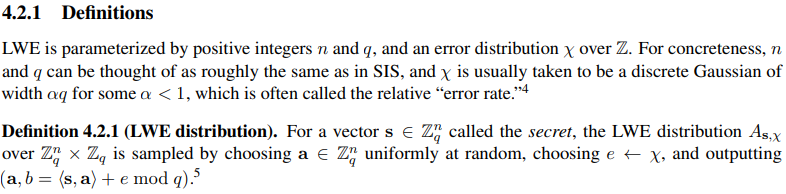
這是前兩個點的gpt



最後一個點



* 1. Learning With Errors (LWE) 學習與錯誤



As,x:LWE distribution，n:維度，q:mod數，α:error rate (α < 1) ，χ是寬度為αq的discrete Gaussian(離散高斯分布)=>可以理解成在正負αq之間根據離散分布隨機取一個數字(越中間機率越高)

LWE distribution的部分s是一個secret向量 ∈ Znq，a是另一個隨機取的向量一樣∈ Znq ， output=(a,b) ， b=< s , a >+e modq∈ Zq ，e是由高斯分布χ隨機取來的。

直覺意義:找個隨機向量，output那個隨機向量和他和s內積的值，為了複雜化問題讓還原變困難 把內積的結果再加上隨機的error。

LWE問題主要有兩個版本:search和decision

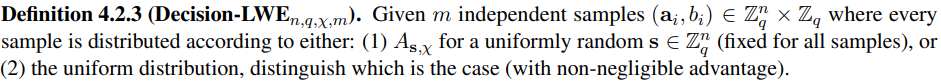
Search version:給一些LWE樣本，找出生成樣本的secret s

Decision version:給一個樣本，判斷是LWE distribution或是random生成的

兩個版本都一樣可以透過m(樣本數)參數化，m足夠大可以確保s唯一



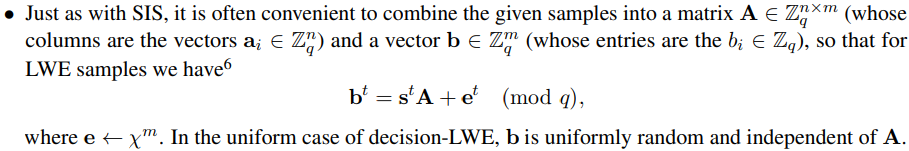
給m個不同的samples( ai , bi ) 要找s。//ai=>隨機向量a， bi=>a和s內積加e



給m個不同的samples，這些samples有兩種可能(1)由s正常生成的(2)隨機生成的。要把這些samples全部分類是那一種

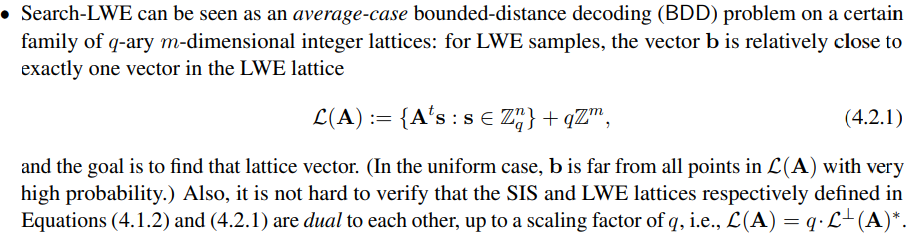
Observations:

1. search和decision版本的問題如果沒有error terms from χ (error e)的話，可以簡單透過Gaussian elimination解決



單項的時候 b=s\*ai+ei =>用矩陣形式表示多項B=sA+E，s是單一向量=>n\*1的matrix。為了符合矩陣格式要對b s e作轉置

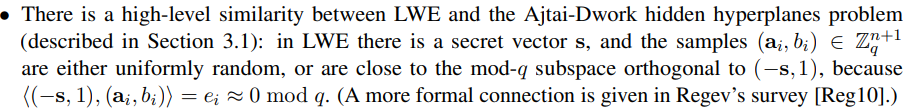
3.



L(A)是所有格子的集合(沒有error項的)，這段在講search lwe中的錯誤項用BDD問題的解法來解，可以想成用很多的BDD把每個向量的error處理掉，然後search lwe就變成比較簡單的問題，可以用矩陣解。

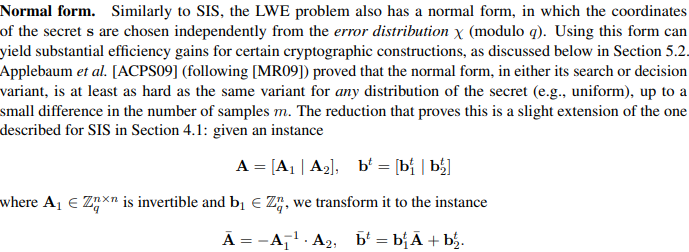
L(A) = q ·L⊥(A) 在講LWE和SIS格子有對偶性，對偶性代表L(A)的向量和L⊥(A)的向量內積都會是整數，可以透過縮放q倍互相轉換。

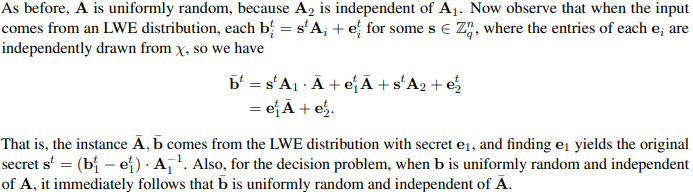
4.

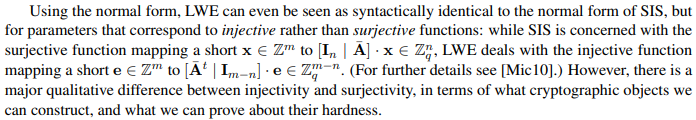


這段的直覺意義是在decision lwe中判斷是否靠近點等同於是隱藏超平面問題判斷是否靠近平面。

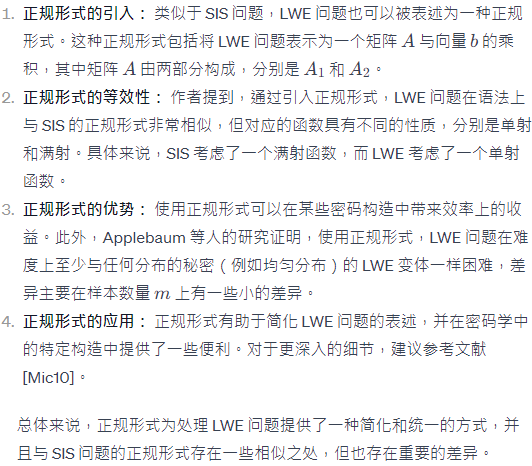
Normal form



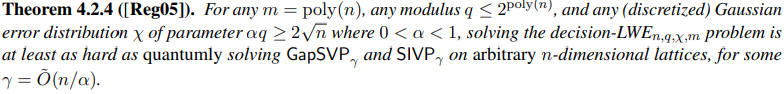




這段normal form的gpt:



4.2.2 Hardness



在上面的這些條件限制上decision lwe至少和一些經典的格問題(GapSVP和SIVP)一樣難

//SIVP是找independent的最短向量組合，GapSVP是確認最短向量是否在某個規定範圍內

1.參數角色(m q α):m和q對LWE的最終難度幾乎沒影響，主要影響的是γ和error rate (1/α) 有關

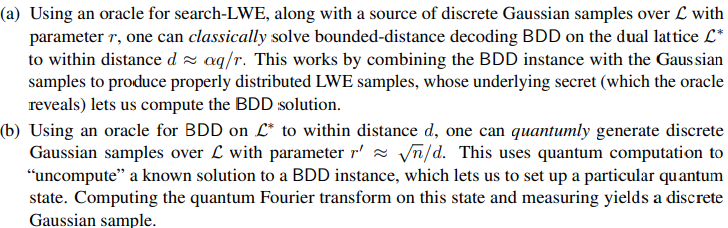
2. quantum polynomial-time reduction:他的證明思路是假設LWE有神奇演算法(oracle)可以解決，那只要展示如何用這個oracle去解GapSVP和SIVP兩個問題。因為這兩個都已知是難問題，表示LWE問題至少和這兩個問題一樣困難

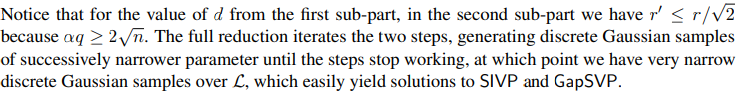
Reduction的概念:A的解法可以拿來解B，代表A難度比B高，可以想成不可能拿簡單問題的解去解難的問題。

1. quantum和classical的reduction證明都有=>可信度高

證明分成兩個部分(search和decision的證明)

1. search lwe

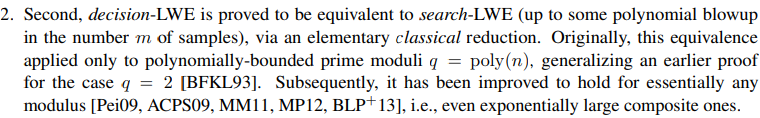




1. 是假設有search lwe的oracle，然後把lwe的參數和BDD的參數搭上關係以後，就可以用oracle去得到BDD的解
2. 是拿(a)步驟的解去生成範圍更窄的discrete gaussian 再拿這個餵給(a)

重複這樣的步驟每輪縮小為1/√ 2倍，最終會得到一個誤差極小的lwe，就可以得到secret就可以用來解GapSVP和SIVP的問題。

1. decision-LWE



Decision的版本是從search版本classical reduction來的，兩個問題的難度等價在任何模數下都成立。

4.2.3 Cryptosystem

公鑰:一些LWE樣本，私鑰:s向量

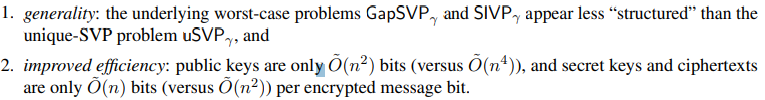
加密1時挑選一些LWE樣本做相加，結果仍然會符合條件(靠近平面)

(因為內積有分配律<a1,s>+<a2,s>=<a1+a2,s>且e1+e2=e3仍會在誤差範圍內)

加密0時會挑一些LWE樣本，也會挑一些隨機樣本，結果會遠離平面

解密透過s去內積a看結果是否接近正確的值來判斷是1還是0。

特點:



1.的structured指得是更結構化，更容易處理和研究

2.public key和secret key和ciphertext大小都減少

4.2.4more hardness還沒看 p27 28 29

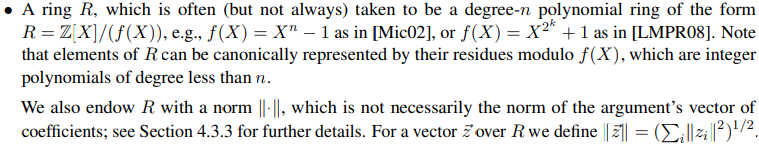
https://blog.csdn.net/forest\_LL/article/details/135895204

* 1. Ring-SIS

前面3.4有提到ring的概念，密鑰大小和運行時間從O(n^2)縮小到O(n)

4.3.1 Definitions

Ring-SIS有一些參數



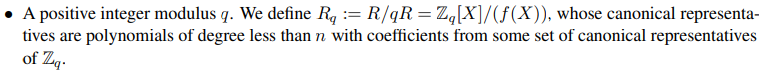
Ring R可以看成是一個degree n的多項式 Z[X]/(f(X))。上面寫道f(X)常用的兩種

從R中隨機挑選一個多項式等同於隨機產生一個不超過n次的多項式

Ring R限制多項式的次數，q限制係數要<q

Ring R可以想成是任意多項式模f(X)的結果，會變一個整數多項式(次數<n)

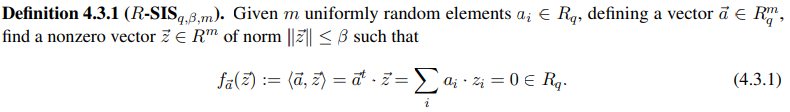
R裡面的每個元素都可以取norm|.|，norm是擴展的係數向量長，取長的方法就是係數的平方和開庚號



可以再把R的係數做大小限制(mod q)



Ring-SIS的關鍵在限定結果的模長，需要一個模的上限β



取m個隨機的多項式ai ∈ Rq，這m個ai組合起來a 向量∈ Rm q (m個Rq)

A可以視為一個m\*n的矩陣 要找到一個z(1\*n)

找到符合長度限制|z向量| ≤ β 的z向量，符合對每一個<ai,z>=0∈ Rq

在ring SIS中ai從Rq取 ，原始版本的ai從Znq取

確保有短整數解的m值會比原始版本的小m ≈ log q，原始SIS需要m ≈ n log q

R-SIS還有優勢是可以用處理多項式(FFT)的技術加速計算

4.3.2 relation to SIS

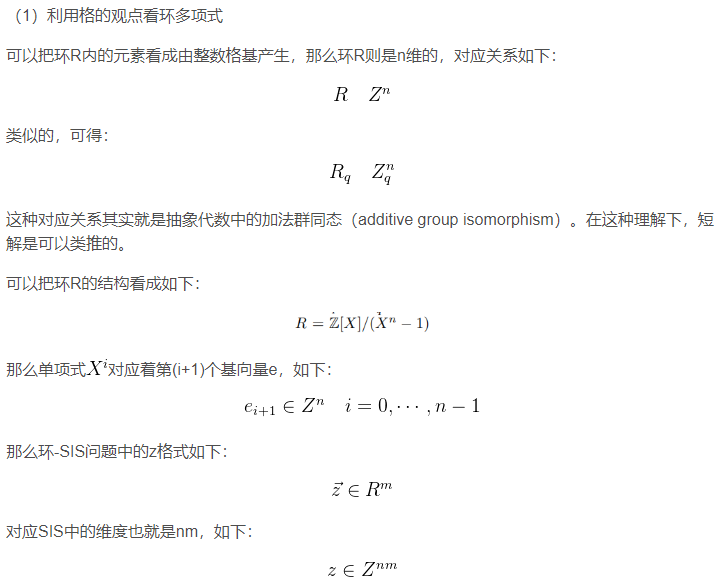
傳統SIS和RING-SIS可以對應

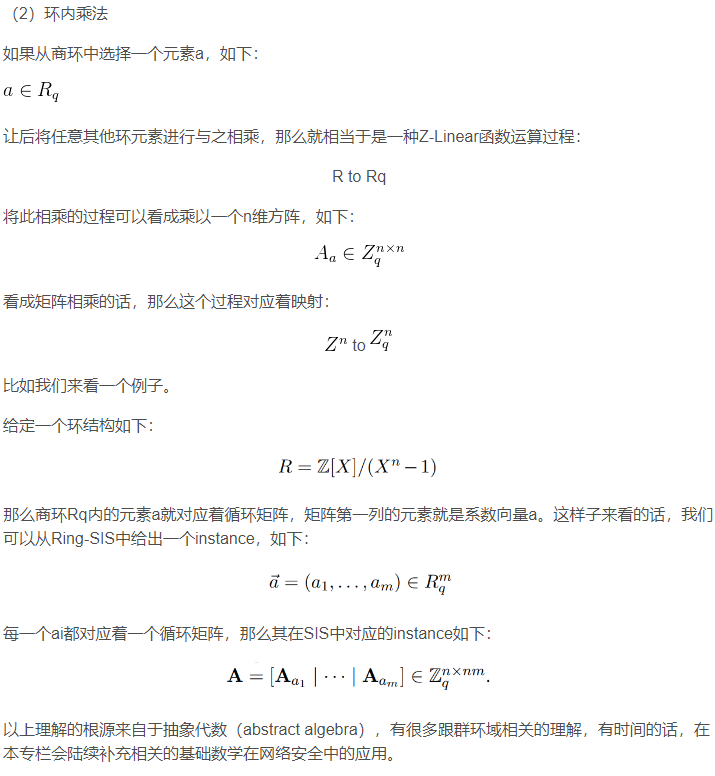
Ring中ai∈ Rq對應到SIS的ai∈ Zn q 可以把n看成在R環上的degree

Ring中的解zi∈ R對應到SIS中的N個整數

可以用格的觀點看環多項式

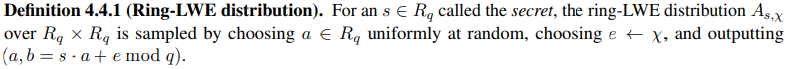
可以把環R內的元素看成是由整數格產生，環R是n維的格子





4.3 沒補完

4.4 Ring LWE



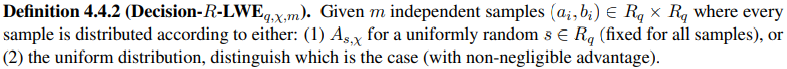
和原始LWE的不同處只有s和a取的來源從Zn q變成商環Rq



t: tweak factor(微調因子) 用來對應tR∨ = R 把R映射到R∨

R∨是R的ideal(對偶理想)，R∨映射的同時保留了原始環R的性質方便計算

在同一次加密時每個b s e用來對應的t是同一個t



和原始版本的decision lwe很多地方是等價的，都可透過樣本數m參數化問題

和原始版本類似，如果沒有error，ring版本的問題也能很簡單解決

Ring lwe的優勢在compactness和effeciency，每個樣本都是一個偽隨機環元素bi∈Rq而不是原版的bi∈Zq 多了環的性質可以用類似FFT的技術來加速

