LWE加解密，代理匹配加密相关介绍

高欣

2024年5月23日

01

LWE加解密

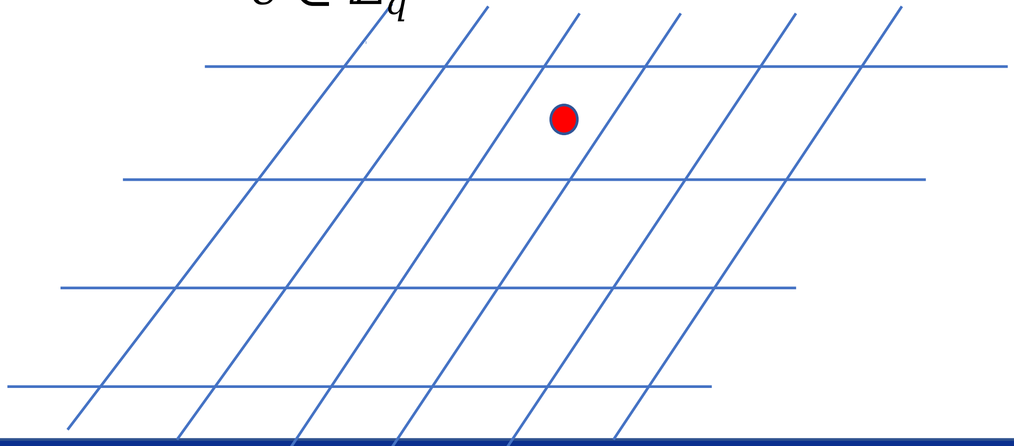
LWE加密

I. 公开矩阵A　　a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$

II. 生成公私钥　　私钥 $s \in \mathbb{Z}_q^m$

公钥（A, b）　　$b = As + e$ ←误差向量

$b \in \mathbb{Z}_q^n$　　$e \in \mathbb{Z}_q^n$



随机$r \in \mathbb{Z}_q^m$

加密 $c_0 = r^t A$， $c_1 = r^t b + \left\lfloor \frac{q}{2} \right\rfloor \mu$ 　　密文：(c_0, c_1)

解密

$$c_1 - c_0 s = r^t (As + e) + \left\lfloor \frac{q}{2} \right\rfloor \mu - r^t As$$

$$= r^t e + \left\lfloor \frac{q}{2} \right\rfloor \mu$$

噪音向量的分布控制的很小

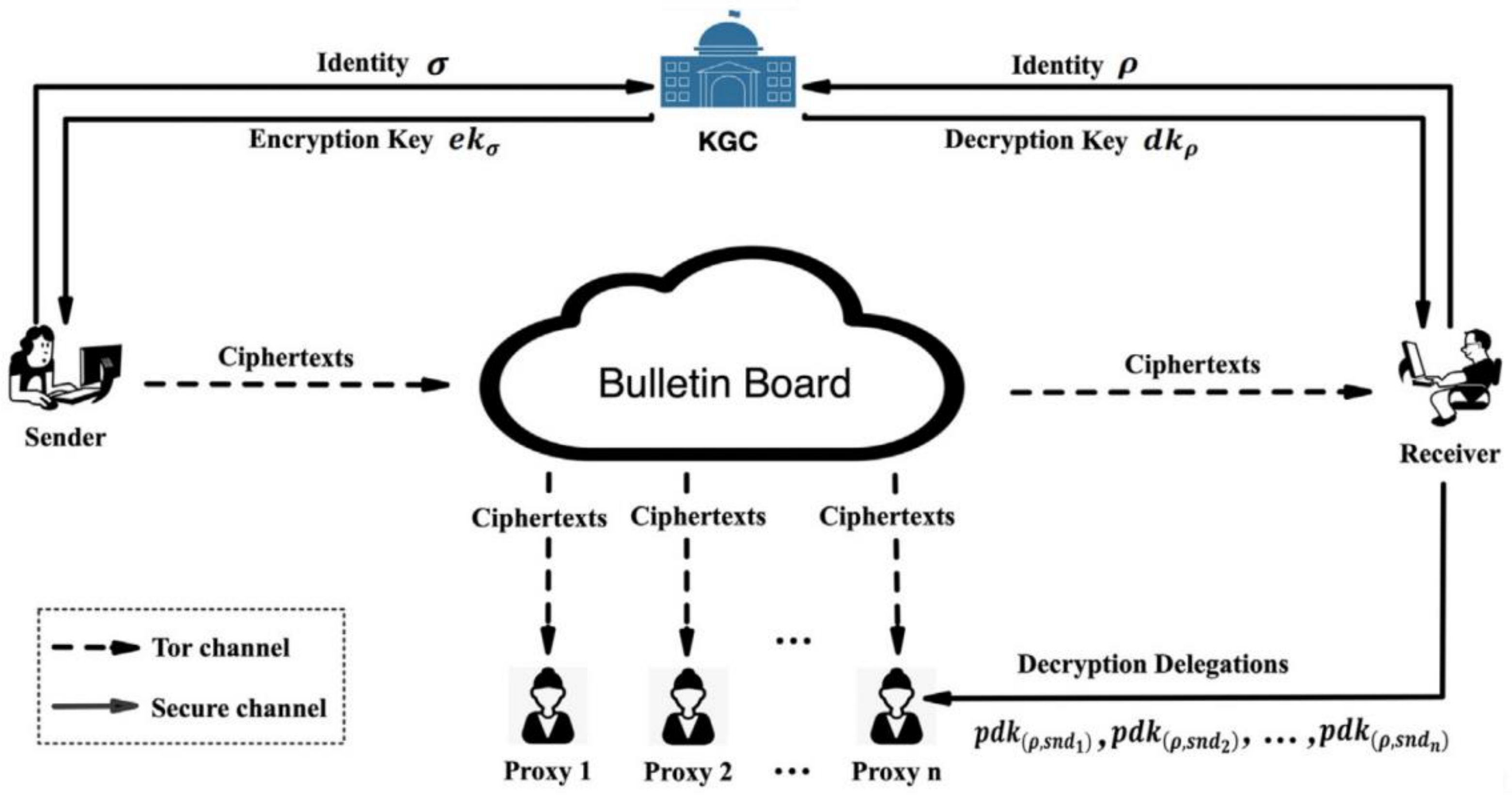
可以直接通过观察结果的值是否小于$\frac{q}{m}$来判断μ是0还是1

02

代理匹配加密

JSA：Identity-based proxy matchmaking encryption for cloud-based anonymous messaging systems

云的匿名消息传递系统

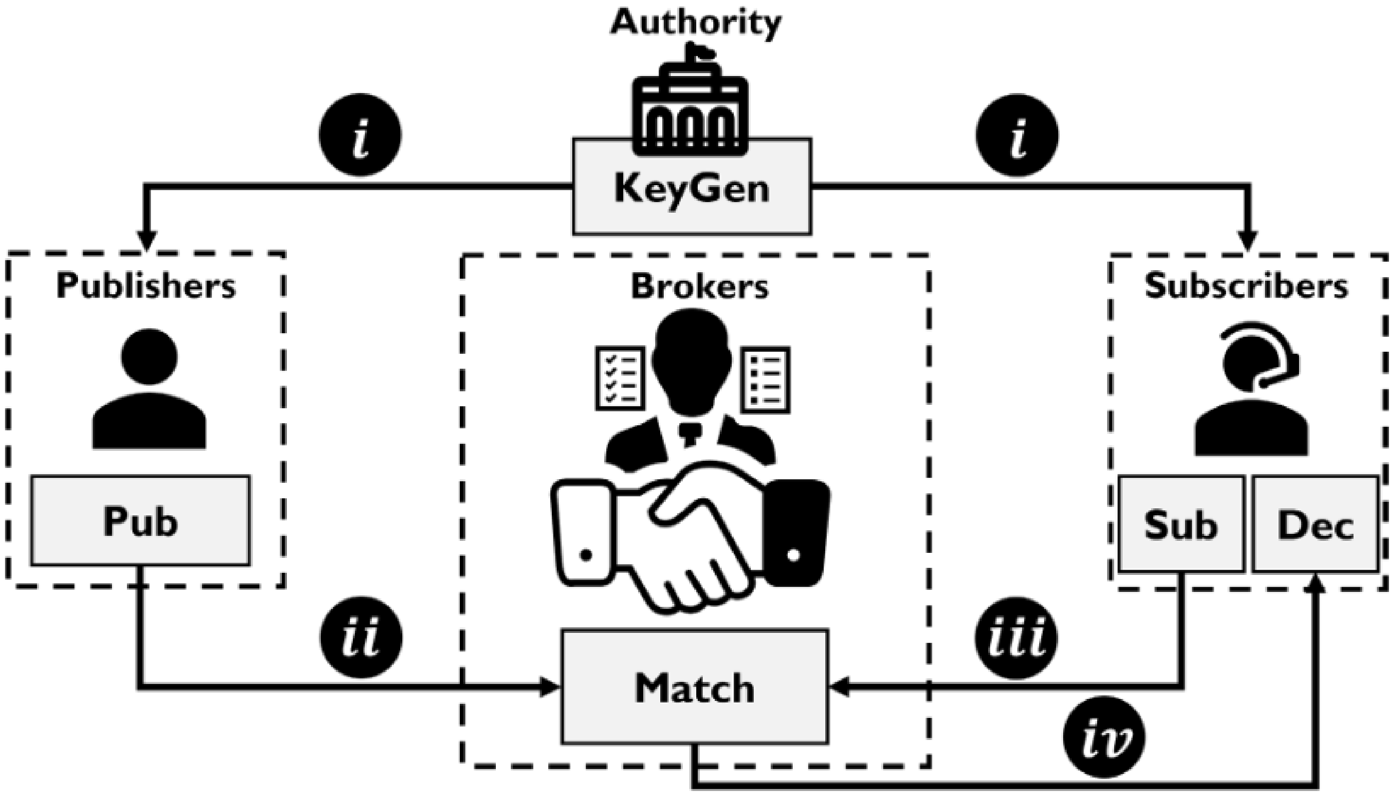


代理：为接收方找到他指定的发送方的密文C，并重加密成CT

- $Setup(\lambda) \rightarrow (pp, mk)$: On input the security parameter λ, this algorithm, run by the KGC, outputs the system parameters pp and the master key mk.
- $SKGen(pp, mk, \sigma) \rightarrow ek_\sigma$: On input the system parameters pp, the master key mk and a sender's identity σ, the encryption key generation algorithm run by the KGC, outputs the corresponding encryption key ek_σ.
- $RKGen(pp, mk, \rho) \rightarrow dk_\rho$: On input the system parameters pp, the master key mk and a receiver's identity ρ, the decryption key generation algorithm run by the KGC, outputs the corresponding decryption key dk_ρ.
- $PKGen(pp, dk_\rho, snd) \rightarrow pdk_{(\rho,snd)}$: On input the system parameters pp, the receiver's decryption key dk_ρ and a target sender's identity snd, the proxy key generation algorithm run by a receiver, outputs the corresponding proxy key $pdk_{(\rho,snd)}$.
- $Enc(pp, ek_\sigma, rcv, m) \rightarrow C$: On input the system parameters pp, the sender's encryption key ek_σ, a target receiver's identity rcv and a message $m \in \mathbb{M}$, the encryption algorithm run by a sender, outputs the corresponding ciphertext C, where $\mathbb{M}$ is the message space.
- $ProxyDec(pp, pdk_{(\rho,snd)}, C) \rightarrow CT/\perp$: On input the system parameters pp, a proxy key $pdk_{(\rho,snd)}$ and a ciphertext C, the proxy decryption algorithm run by a proxy, outputs the corresponding proxy ciphertext CT or a symbol $\perp$ to denote proxy decryption failure.
- $Dec_1(pp, dk_\rho, snd, C) \rightarrow m/\perp$: On input the system parameters pp, the receiver's decryption key dk_ρ, a target sender's identity snd and a ciphertext C, the algorithm run by a receiver, outputs the corresponding message m or a symbol $\perp$ to denote decryption failure.
- $Dec_2(pp, dk_\rho, snd, CT) \rightarrow m/\perp$: On input the system parameters pp, the receiver's decryption key dk_ρ, a target sender's identity snd and a proxy ciphertext CT, the decryption algorithm run by a receiver, outputs the corresponding message m or a symbol $\perp$ to denote decryption failure.

Pub/Sub（发布/订阅）是一种消息传递模式，其中消息发送者（发布者）将消息发布到一个或多个主题（topics）或频道（channels），而消息接收者（订阅者）订阅特定的主题或频道以接收消息。

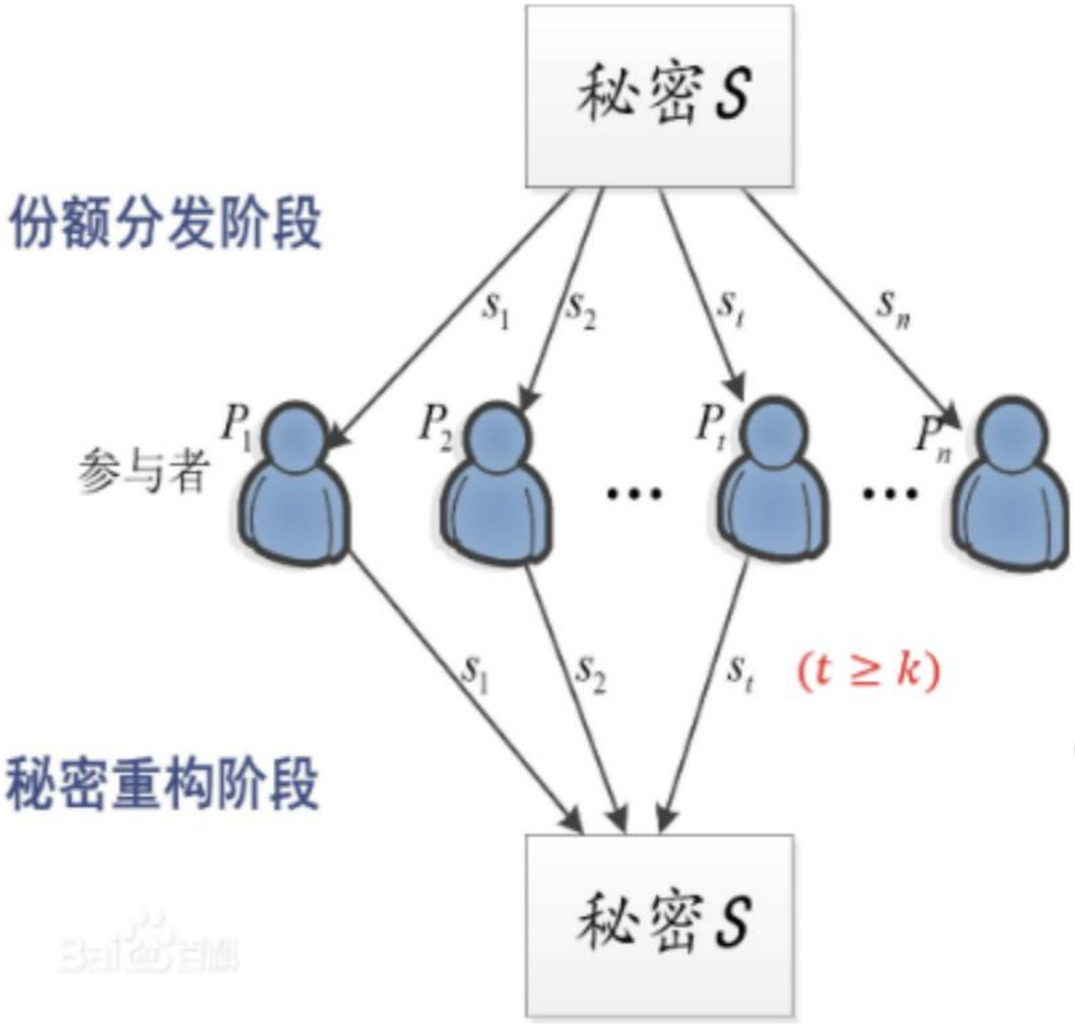
在Pub/Sub模式中，发布者和订阅者不直接通信，而是通过一个中介（通常称为消息代理或消息中间件）进行通信。发布者将消息发送到消息代理，消息代理将消息存储在某个地方，并根据订阅者的订阅列表将消息推送给相应的订阅者。



TIFS：Secure Cloud-Assisted Data Pub/Sub Service With Fine-Grained Bilateral Access Control

03

Shamir秘密共享



(k,n)秘密分割门限方案，k为门限值

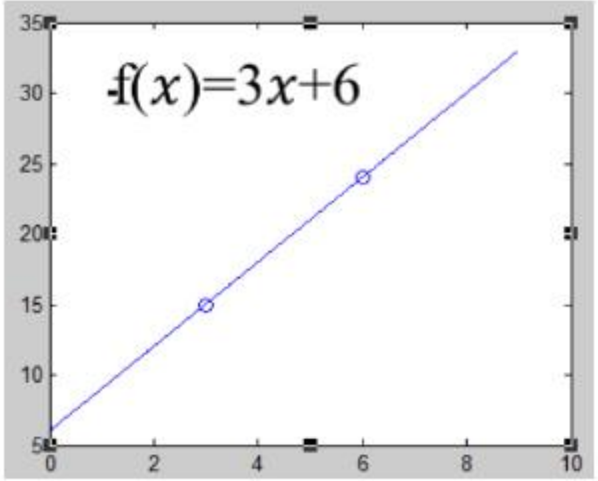
秘密s 被分为n个部分,每个部分称为份额share或shadow,由一个参与者持有，使得：

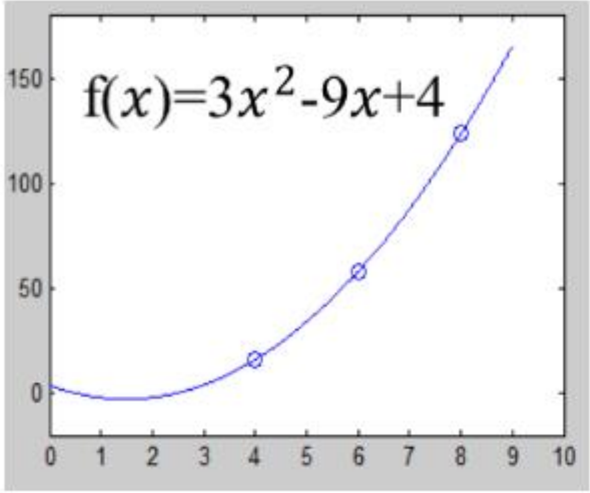
- 由k个或多于k个参与者所持有的部分信息可重构s;
- 由少于k个参与者所持有的部分信息则无法重构s

$$(|S_A \cap P_A| \geq d) \cap (|S_B \cap P_B| \geq d)),$$

阈值访问控制

□ Shamir门限方案的构造思路





一般的，设$\{(\boldsymbol{x_1}, \boldsymbol{y_1}), \ldots, (\boldsymbol{x_k}, \boldsymbol{y_k})\}$是平面上$\boldsymbol{k}$个不同的点构成的点集，那么在平面上存在唯一的$\boldsymbol{k}-\boldsymbol{1}$次多项式$\boldsymbol{f}(\boldsymbol{x}) = a_0 + a_1 x + \cdots + a_{k-1}x^{k-1}$通过这$\boldsymbol{k}$个点.

若把秘密$\boldsymbol{s}$取做$\boldsymbol{f}(\boldsymbol{0})$，$\boldsymbol{n}$个份额取做 $\boldsymbol{f}(\boldsymbol{i})$ $(\boldsymbol{i} = \boldsymbol{1}, \ldots \boldsymbol{n})$，那么利用其中任意$\boldsymbol{k}$个份额可以重构$\boldsymbol{f}(\boldsymbol{x})$，从而可以得到秘密$\boldsymbol{s}$.

FIBE

Fuzzy.Setup$(1^\lambda, 1^\ell)$: On input a security parameter λ, and identity size ℓ, do:

1. Use algorithm $\mathsf{TrapGen}(1^\lambda)$ (from Proposition 3) to select 2ℓ uniformly random $n \times m$-matrices $\mathbf{A}_{i,b} \in \mathbb{Z}_q^{n\times m}$ (for all $i \in [\ell], b \in \{0,1\}$) together with a full-rank set of vectors $\mathbf{T}_{i,b} \subseteq \Lambda_q^{\perp}(\mathbf{A}_{i,b})$ such that $\|\widetilde{\mathbf{T}_{i,b}}\| \leq m \cdot \omega(\sqrt{\log m})$.
2. Select a uniformly random vector $\mathbf{u} \in \mathbb{Z}_q^n$.
3. Output the public parameters and master key,

$$\mathsf{PP} = \left(\{\mathbf{A}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}, \mathbf{u} \right) \quad ; \quad \mathsf{MK} = \left(\{\mathbf{T}_{i,b}\}_{i\in[\ell],b\in\{0,1\}} \right)$$

拉格朗日插值定理 $L_n(x) = \sum_{j=0}^{n-1} y_j p_j(x)$

有n个互不相同的点$(x_1, y_1), \ldots, (x_n, y_n)$，存在唯一的n-1次多项式经过这n个点

$\mathsf{Fuzzy.Extract}(\mathsf{PP}, \mathsf{MK}, \mathsf{id}, k)$**:** On input public parameters PP, a master key MK, an identity $\mathsf{id} \in \{0,1\}^\ell$ and threshold $k \leq \ell$, do:

1. Construct ℓ shares of $\mathbf{u} = (u_1, \ldots, u_n) \in \mathbb{Z}_q^n$ using a Shamir secret-sharing scheme applied to each co-ordinate of $\mathbf{u}$ independently. Namely, for each $j \in [n]$, choose a uniformly random polynomial $p_j \in \mathbb{Z}_q[x]$ of degree $k-1$ such that $p_j(0) = u_j$.
   Construct the j^{th} share vector

$$\hat{\mathbf{u}}_j = (\hat{u}_{j,1}, \ldots, \hat{u}_{j,n}) \stackrel{\text{def}}{=} (p_1(j), p_2(j), \ldots, p_n(j)) \in \mathbb{Z}_q^n$$

   Looking ahead (to decryption), note that for all $J \subset [\ell]$ such that $|J| \geq k$, we can compute fractional Lagrangian coefficients L_j such that $\mathbf{u} = \sum_{j \in J} L_j \cdot \hat{\mathbf{u}}_j \pmod q$. That is, we interpret L_j as a fraction of integers, which we can also evaluate $\pmod q$.

2. Using trapdoor MK and the algorithm $\mathsf{SamplePre}$ from Section 3.3.1, find $\mathbf{e}_j \in \mathbb{Z}^m$ such that $\mathbf{A}_{j,\mathsf{id}_j} \cdot \mathbf{e}_j = \hat{\mathbf{u}}_j$, for $j \in [\ell]$.
3. Output the secret key for id as $(\mathbf{e}_1, \ldots, \mathbf{e}_\ell)$.

加密矩阵大小恒定

Fuzzy.Setup$(1^\lambda, 1^\ell)$**:** On input a security parameter λ, and identity size ℓ, do these steps:

1. Select a uniformly random n-vector $\mathbf{u} \xleftarrow{R} \mathbb{Z}_q^n$.
2. For $(i = 1, \ldots, \ell)$
   (a) Use algorithm $\mathsf{TrapGen}(q, n)$ to select a uniformly random $n \times m$-matrix $\mathbf{A}_{0,i} \in \mathbb{Z}_q^{n\times m}$ with a basis $\mathbf{T}_{\mathbf{A}_{0,i}}$ for $\Lambda_q^\perp(\mathbf{A}_{0,i})$ such that $\|\widetilde{T_{\mathbf{A}_{0,i}}}\| \leq O(\sqrt{n \log q})$
   (b) Select two uniformly random $n \times m$ matrices $\mathbf{A}_{1,i}$ and $\mathbf{B}_i$ in $\mathbb{Z}_q^{n\times m}$.
3. Output the public parameters and master key,

$$\mathsf{PP} = \left(\{\mathbf{A}_{0,i}, \mathbf{A}_{1,i}, \mathbf{B}_i\}_{i\in[\ell]}, \quad \mathbf{u} \right) \quad ; \quad \mathsf{MK} = \left(\{T_{\mathbf{A}_{0,i}}\}_{i\in[\ell]} \right)$$

Fuzzy.Enc$(\mathsf{PP}, \mathsf{id}, b)$**:** On input public parameters PP, an identity id, and a message $b \in \{0,1\}$, do:

1. Let $D \stackrel{\text{def}}{=} (\ell!)^2$.
2. Choose a uniformly random $\mathbf{s} \stackrel{R}{\leftarrow} \mathbb{Z}_q^n$.
3. Choose a noise term $x \leftarrow \chi_{\{\alpha,q\}}$ and $\mathbf{x}_i \leftarrow {\chi_{\{\alpha,q\}}}^m$,
4. Set $c_0 \leftarrow \mathbf{u}^\top \mathbf{s} + Dx + b\lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$.
5. Set $\mathbf{c}_i \leftarrow {\mathbf{A}_{i,\mathsf{id}_i}}^\top \mathbf{s} + D\mathbf{x}_i \in \mathbb{Z}_q^m$ for all $i \in [\ell]$.
6. Output the ciphertext $\mathsf{CT}_{\mathsf{id}} := (c_0, \{\mathbf{c}_i\}_{i\in[\ell]})$.

$$\sum_{j\in J} L_j \mathbf{A}_j \mathbf{e}_j = \mathbf{u} \pmod q$$

$$\begin{aligned}
r &= c_0 - \sum_{j\in J} L_j\, \mathbf{e}_j^\top\, \mathbf{c}_j \pmod q \\
&= \mathbf{u}^\top \mathbf{s} + Dx + b\left\lfloor \frac{q}{2} \right\rfloor - \sum_{j\in J} L_j\, \mathbf{e}_j^\top ({\mathbf{A}_j}^\top \mathbf{s} + D \cdot \mathbf{x}_j) \pmod q \\
&= b\left\lfloor \frac{q}{2} \right\rfloor + \underbrace{\left(\mathbf{u}^\top \mathbf{s} - \sum_{j\in J} (L_j\, \mathbf{A}_j \mathbf{e}_j)^\top \mathbf{s}\right)}_{=0 \pmod q} + \underbrace{\left(Dx - \sum_{j\in J} D L_j \mathbf{e}_j^\top \mathbf{x}_j\right)}_{\approx 0} \pmod q \qquad \approx b\left\lfloor \frac{q}{2} \right\rfloor
\end{aligned}$$

8. If $|r| < \frac{q}{4}$, output 0, else output 1.

利用SIS单向函数的反函数构造

Fuzzy.Extract$(\mathsf{PP}, \mathsf{MK}, \mathsf{id}, k)$**:** On input public parameters PP, a master key MK, an attribute vector or identity $\mathsf{id} = (\mathsf{id}_1, \mathsf{id}_2, \ldots, \mathsf{id}_\ell)$ where $\mathsf{id}_i \in \mathbb{Z}_q^n$ for each $i \in [\ell]$, and a threshold $k \le \ell$, do:

1. Construct ℓ shares of $\mathbf{u} = (u_1, ..., u_n) \in \mathbb{Z}_q^n$ using a Shamir secret-sharing scheme applied to each co-ordinate of $\mathbf{u}$ independently. Namely, for each $i \in [n]$, choose a uniformly random polynomial $p_i \in \mathbb{Z}_q[x]$ of degree $k-1$ such that $p_i(0) = u_j$.
2. Construct the j^{th} share vector

$$\hat{\mathbf{u}}_j = (\hat{u}_{j,1}, \ldots, \hat{u}_{j,n}) \stackrel{\text{def}}{=} (p_1(j), p_2(j), \ldots, p_n(j)) \in \mathbb{Z}_q^n$$

   Note that by the linearity of the Shamir secret-sharing scheme, there are co-efficients $L_j \in \mathbb{Z}_q$ such that $\mathbf{u} = \sum_{j=1}^{\ell} L_j \cdot \hat{\mathbf{u}}_j$. In fact, linear reconstruction is possible whenever there are k or more shares available.
3. For $i = 1, \ldots, \ell$, do:
   (a) For id_i, construct the *encryption* matrix $\boxed{\mathbf{F}_{\mathsf{id}_i} = [\mathbf{A}_{0,i} | \mathbf{A}_{1,i} + \mathbf{H}(\mathsf{id}_i)\mathbf{B}_i}]$ as in [1]. Here, $\mathbf{H}$ is some fixed Full-Rank Difference (FRD) map, s.t., for any $\mathsf{id}_1 \neq \mathsf{id}_2$ in some exponential-size domain, $\mathbf{H}(\mathsf{id}_1) - \mathbf{H}(\mathsf{id}_2)$ is a full-rank matrix.
   (b) Sample $\mathbf{e}_i \in \mathbb{Z}^{2m}$ as $\mathbf{e}_i \leftarrow \mathsf{SampleLeft}(\mathbf{A}_{0,i}, \quad \mathbf{A}_{1,i} + \mathbf{H}(\mathsf{id}_i)\,\mathbf{B}_i, \quad \mathbf{T}_{\mathbf{A}_{0,i}}, \quad \hat{\mathbf{u}}, \quad \sigma)$
4. Output the secret key $\mathsf{SK}_{\mathsf{id}} = (\mathbf{e}_1, \ldots, \mathbf{e}_\ell)$.

$\mathsf{Fuzzy.Enc}(\mathsf{PP}, \mathsf{id}, b)$**:** On input PP, identity $\mathsf{id} = (\mathsf{id}_1, \mathsf{id}_2, \ldots, \mathsf{id}_\ell) \in (\mathbb{Z}_q^n)^\ell$, and message $b \in \{0, 1\}$:

1. Let $D = (\ell!)^2$.
2. Choose a uniformly random $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n$.
3. For $(i = 1, \ldots, \ell)$, do:
   (a) Construct the *encryption* matrix $\mathbf{F}_{\mathsf{id}_i} = [\mathbf{A}_{0,i} | \mathbf{A}_{1,i} + \mathbf{H}(\mathsf{id}_i)\mathbf{B}_i] \in \mathbb{Z}_q^{n \times m}$ as above.
   (b) Choose a uniformly random $m \times m$ matrix $R \xleftarrow{R} \{-1, 1\}^{m \times m}$.
   (c) Choose noise vector $y \xleftarrow{\bar{\Psi}_\alpha^m} \mathbb{Z}_q^m$, and set $z \leftarrow R^\top y \in \mathbb{Z}_q^m$.
   (d) Set $\mathbf{c}_i \leftarrow \mathbf{F}_{\mathsf{id}_i}{}^\top \mathbf{s} + D \begin{bmatrix} y \\ z \end{bmatrix} \in \mathbb{Z}_q^{2m}$ for all $i \in [\ell]$.
4. Choose a noise term $x \xleftarrow{\chi_{\{\alpha, q\}}} \mathbb{Z}_q$.
5. Set $\mathbf{c}_0 \leftarrow \mathbf{u}^\top \mathbf{s} + Dx + b \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$.
6. Output the ciphertext $\mathsf{CT}_{\mathsf{id}} := (c_0, \{\mathbf{c}_i\}_{i \in [\ell]})$.

$\mathsf{Fuzzy.Dec}(\mathsf{PP}, \mathsf{SK}_{\mathsf{id}}, \mathsf{CT}_{\mathsf{id}'})$**:** On input parameters PP, a private key $\mathsf{SK}_{\mathsf{id}}$, and a ciphertext $\mathsf{CT}_{\mathsf{id}'}$:

1. Let $J \subset [\ell]$ denote the set of matching elements in id and id'. If $|J| \geq k$ we can compute Lagrange coefficients L_j so that

$$\sum_{j \in J} L_j \hat{\mathbf{u}}_j = \sum_{j \in J} L_j \mathbf{F}_{\mathsf{id}_j}{}^\top \mathbf{e}_j = \mathbf{u}$$

2. Compute $r \leftarrow c_0 - \sum_{j \in J} L_j \cdot \mathbf{e}_j^\top \mathbf{c}_j \pmod q$. View it as the integer $r \in [-\lfloor \frac{q}{2} \rfloor, \lfloor \frac{q}{2} \rfloor) \subset \mathbb{Z}$.
3. If $|r| < \frac{q}{4}$, output 0, else output 1.