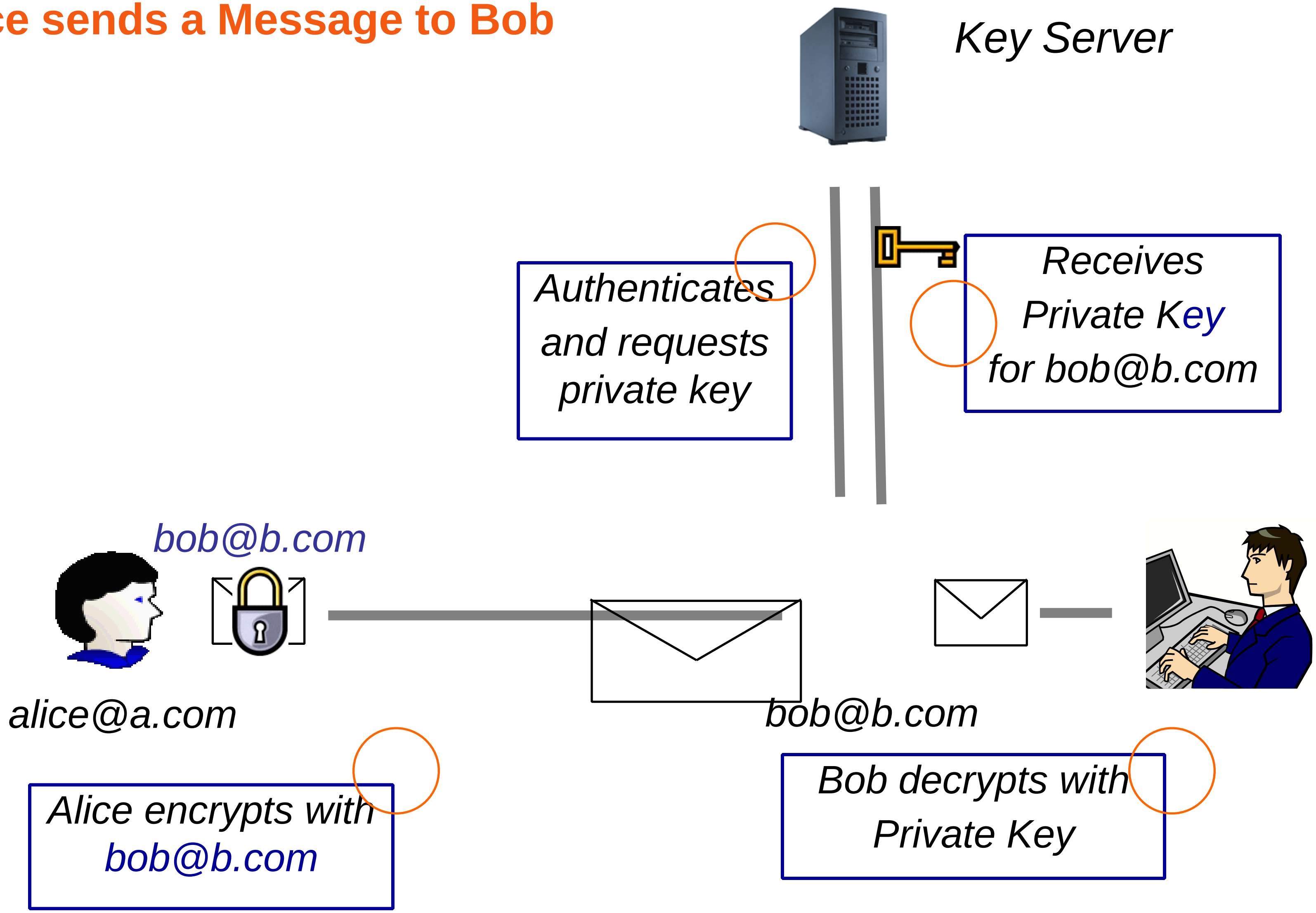
Lattice (H) IBE in the standard model with short public parameter

Kunwar Singh

NIT Trichy

How IBE works in practice

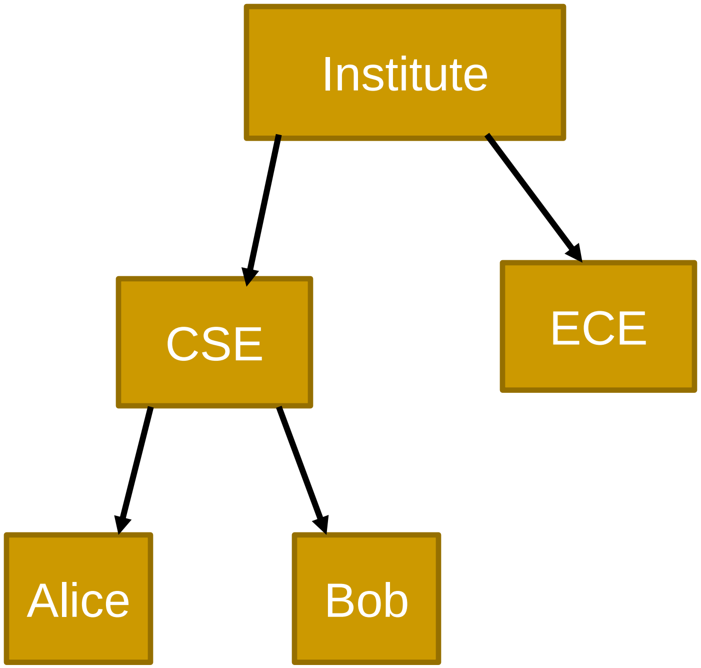
Alice sends a Message to Bob



Hierarchical IBE (HIBE)

HIBE primitive [HL02, GS02]

- PKG (root) delegates the capability for providing private key generation and identity authentication to lower level entities.
- is the hierarchical extension of IBE schemes.
- There are no lower level public parameters. Only the PKG has public parameters.
- Alice can obtain her private key from her "local" key generation centre.



CSE : Lower level KGC

ID_{Alice} = (Institute,CSE,Alice)

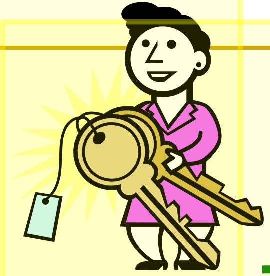
HIBE Scheme

- Setup(λ,d): Outputs PP and MSK.
- Derive(ID, $ID_{\text{prefix of ID-ID}}$,PP): Outputs d_{ID} .
- Encrypt(ID_R ,m,PP): Outputs ciphertext C.
- Decrypt(C, d_{IDr},PP): Outputs m.

IBE: is special case of HIBE when depth is one.

Game!

Get instance of hard problem H

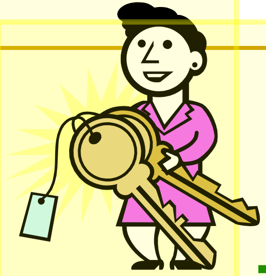


Construct prams from H

Challenger

Adversary

Get instance of
hard problem H



Game!

Construct prams from H

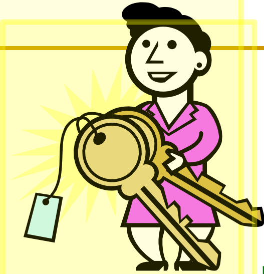
ID1,ID2,…,IDm

$d_{ID1}, d_{ID2}, \ldots, d_{IDm},$

Challenger

Adversary

Get instance of hard problem H



Game!

Construct prams from H

ID1,ID2,…,IDm

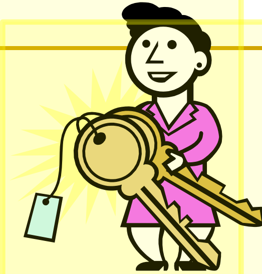
$d_{ID1}, d_{ID2}, \ldots , d_{IDm},$

$C^{*}= C_{random}$ OR $\mathrm{Enc}(\mathrm{m}, ID^{*}, \mathrm{params})$

Challenger

Adversary

Get instance of hard problem H



Game!

Challenger

Adversary

Construct prams from H

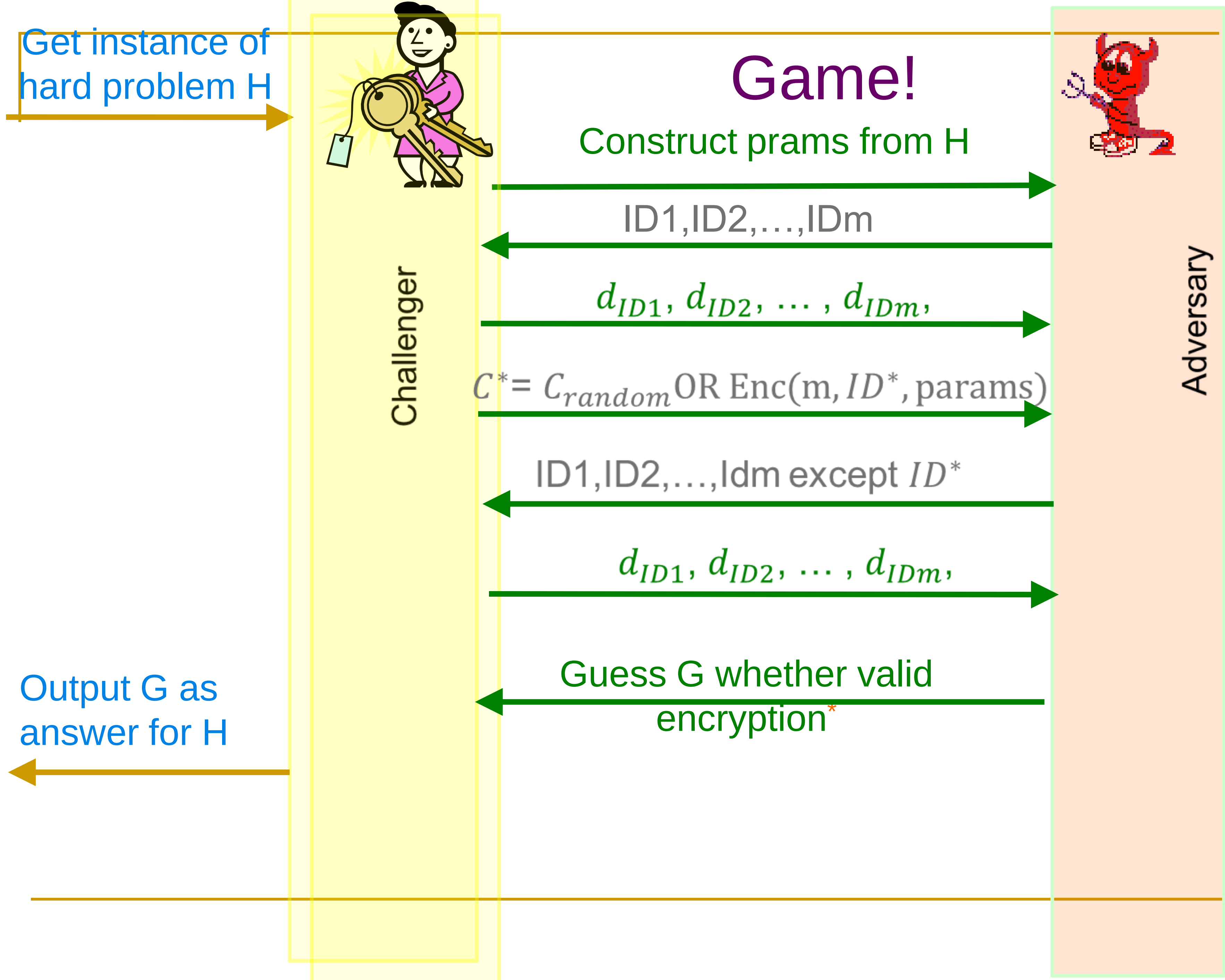
ID1,ID2,…,IDm

$d_{ID1}, d_{ID2}, \ldots , d_{IDm},$

$C^{*}= C_{random}$ OR Enc(m, ID^{*}, params)

ID1,ID2,…,Idm except ID^{*}

$d_{ID1}, d_{ID2}, \ldots , d_{IDm},$


Get instance of hard problem H
Game!
Construct prams from H
ID1,ID2,…,IDm
$d_{ID1}, d_{ID2}, \dots, d_{IDm},$
$C^* = C_{random}$ OR Enc(m, ID^*, params)
ID1,ID2,…,Idm except ID^*
$d_{ID1}, d_{ID2}, \dots, d_{IDm},$
Guess G whether valid encryption*
Output G as answer for H
Challenger
Adversary

Lattices

Motivations:

- Alternating option
- Strong hardness guarantees
- Efficient operations, parallelizable
- No quantum algorithm (yet)
- Fully Homomorphic Encryption (Secure Computation)

Lattices

Definition : A Lattice is set of integer linear combination of n linearly independent vectors.

$$L=\{b_1x_1+\ldots+b_nx_n \mid x_i \text{ integers}\}$$

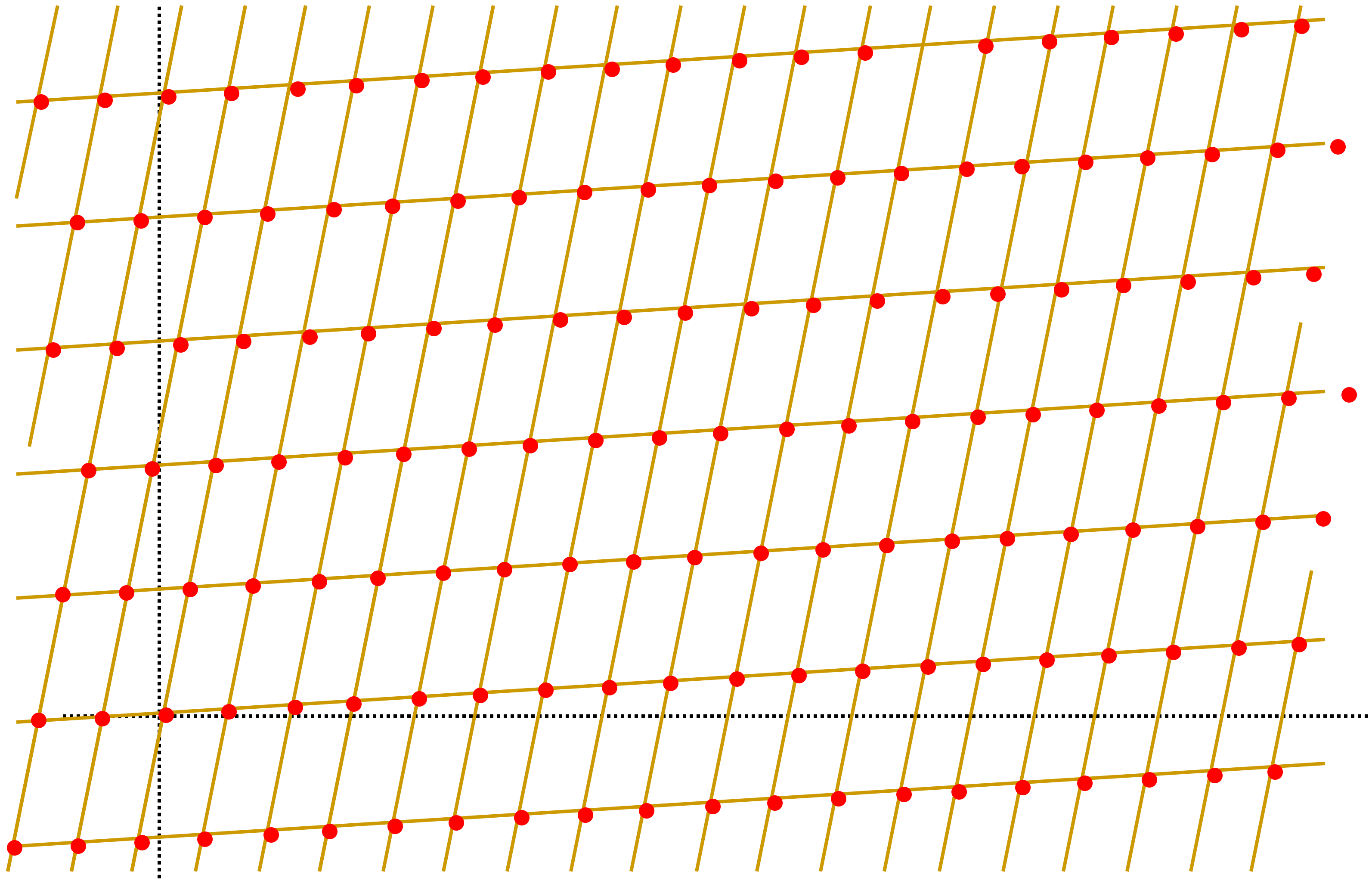
The vectors $b_1,\ldots,b_n$ are a bases for L.

Equivalently, a lattice L is a set of points in n-dimension with periodic structure.

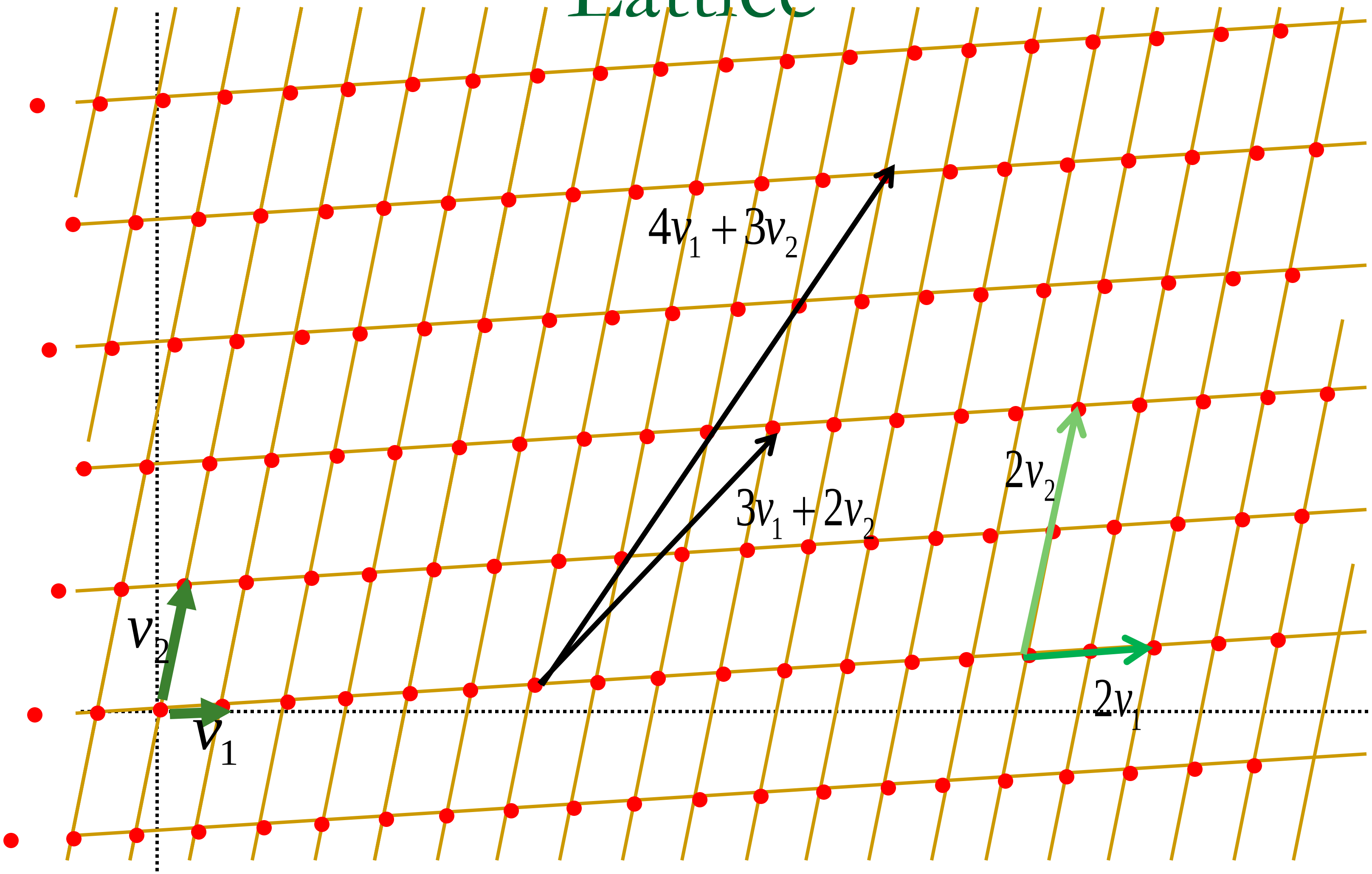
- Rank is number of independent vectors. Dimension is size of vector
- Full rank lattice, Rank = Dimension
- Lattices are represented by bases. Bases are not unique, but they can be obtained from each other by integer transforms of determinant ±1. Each Lattice has many basis.

$$\begin{bmatrix} 3 & 4 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 4 \\ 2 & 3 \end{bmatrix}$$

Lattice



Lattice



Hard Problems in Lattice

- Shortest vector Problem (SVP): Find a shortest nonzero vector in lattice.
- Closest vector problem (CVP): Given a vector

w$\in R^n$ that is not in L. Find a vector v$\in$ L that is closest to w.

Babai's closest vertex algorithm: Let $L \subset R^n$ has a basis $v_1,\ldots,$ v_n and let $w \in R^n$ be an arbitrary vector.

Write $w = t_1 v_1 + \ldots + t_n v_n$ with $t_1, \ldots, t_n \in R$

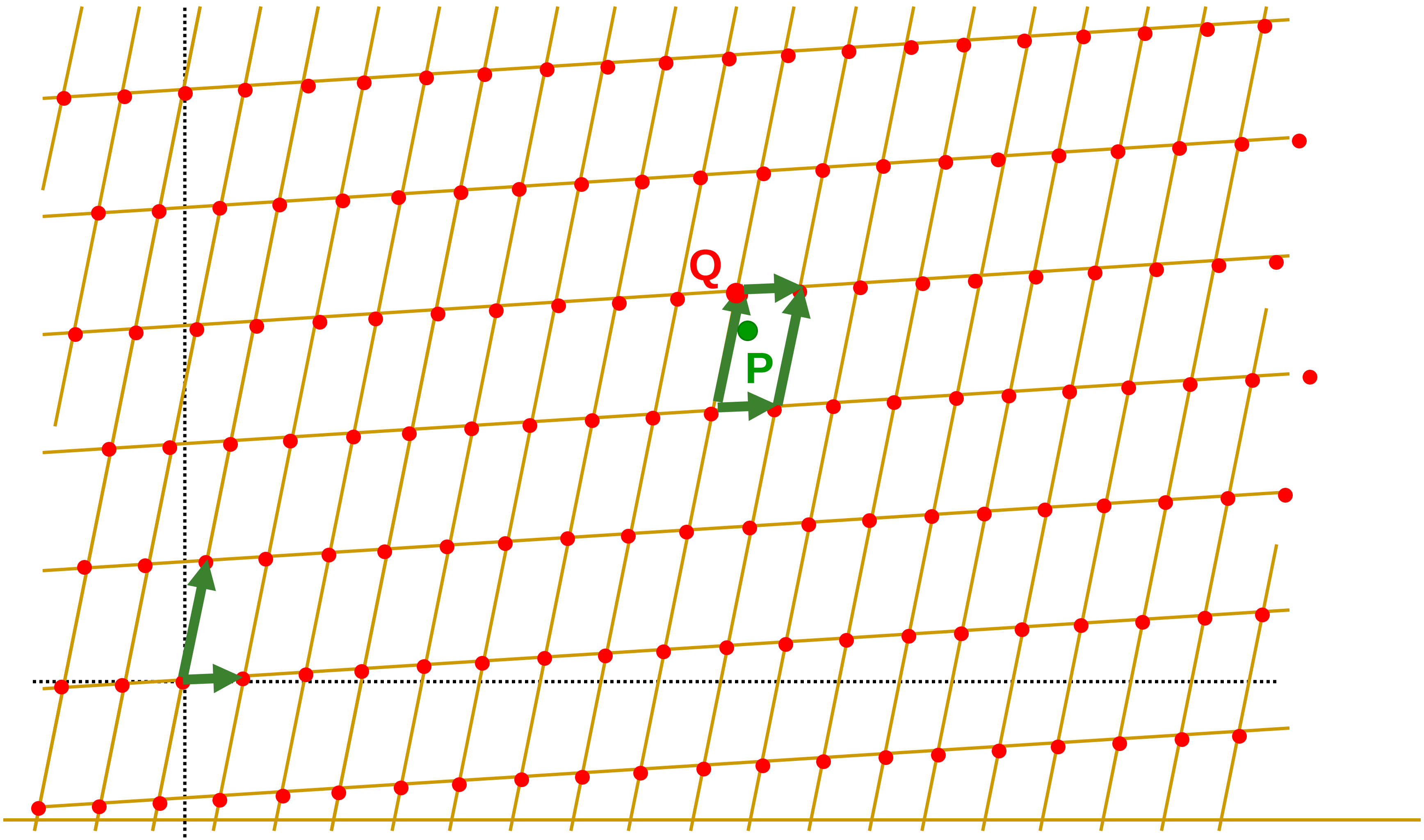
Set $a_i = [t_i]$ for $i = 1,\ldots,n$.

Return the vector $a_1 v_1 + \ldots + a_n v_n$.

- If the vectors in the basis are reasonably orthogonal to one another, then the algorithm solves some version of apprCVP .
- If basis are highly nonorthogonal, then the vector returned by algorithm is generally far from the closest lattice vector.

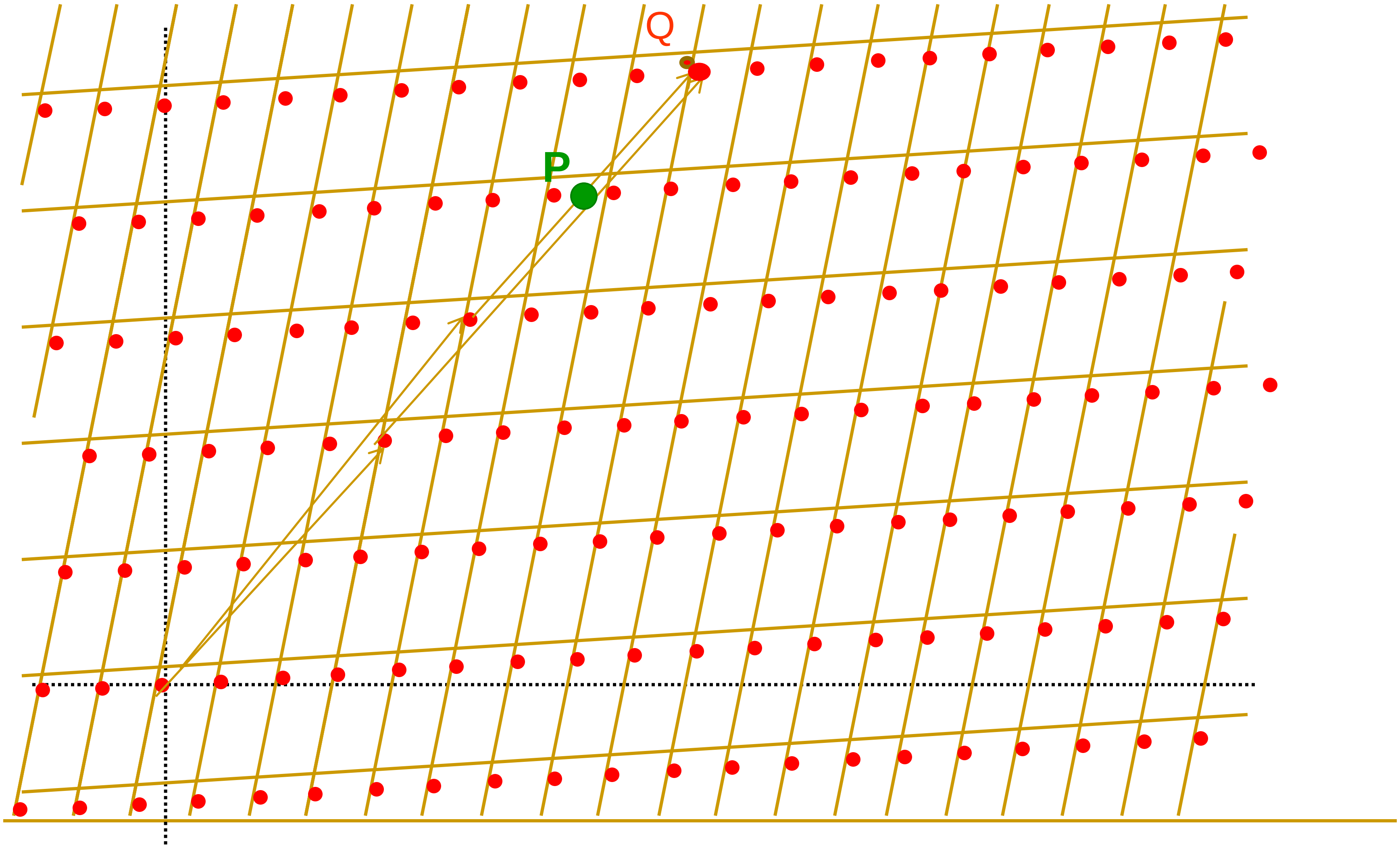
Closest vector problem

Good basis=Orthogonal basis = Short basis



Shortest vector problem

Bad basis = highly nonorthogonal basis



So cryptosystem based on lattice

- Make bad basis public key
- Make good basis private key
- Encrypt using bad basis, decrypt using good basis
- Recovering good basis from bad basis is hard !

Regev' Learning With Error (LWE)Problem

Search: Given an arbitrary number of 'approximate' random linear equation on s $\in Z_{17}^4$.

$$14s_1 + 15s_2 + 5s_3 + 2s_4 \approx 8 (\mathrm{mod} 17)$$

$$13s_1 + 14s_2 + 14s_3 + 6s_4 \approx 16 (\mathrm{mod} 17)$$

$$6s_1 + 10s_2 + 13s_3 + 1s_4 \approx 3 (\mathrm{mod} 17)$$

$$\vdots \qquad\qquad \vdots$$

$$\vdots \qquad\qquad \vdots$$

Find: s $\in Z_q^n$ is hard, when n ≥ 500, q is polynomial in n.

More precisely:

- Fix a size parameter n ≥ 1, a modulus q ≥ 2 and an ‘error’ probability ditribution (Gaussian) χ on Z.
- An oracle (who knows s) generates a uniform vector $a \in Z_q^n$ and noise $e \in Z$ according to χ.
- The Oracle outputs (a ,$a\langle a, s\rangle + e$).
- This procedure is repeated arbitrary number of times with s and fresh a and e .
- Find s is hard.

Decision version:

Distinguish between following two oracles:

Oracle 1: Outputs samples of the form $(a, \langle a, s\rangle + e)$, where s is fixed, a is uniform in Z_q^n and $e \in Z$ is fresh sample from χ.

Oracle 2: Outputs truly uniform samples from $Z_q^n \times Z_q$.

- The Small Integer Solution (SIS) problem: Given an integer q, a matrix $A \in Z_q^{n\times m}$, a real β,find a nonzero integer vector $e \in Z^m$ such that $Ae = 0$ mod q and $\|e\| \leq \beta$.
- The Inhomogeneous Small Integer Solution (ISIS) problem: Given an integer q, syndrome u, a matrix $A \in Z_q^{n\times m}$,a real β,find a nonzero integer vector $e \in Z^m$ such that $Ae = u$ mod q and $\|e\| \leq \beta$.

LWE based Public Key Cryptosystem

- System Parameter: Integers n (the security parameter), m (number of equations), q modulus, and a real α > 0 (noise parameter).
- Private Key: is a vector $s \in_R Z_q^n$
- Public Key: consists of m samples $(a_i, b_i)_{i=1}^m$ from the LWE distribution with secret s. $a_i \in_R Z_q^n$ and $b_i \in Z_q$. OR $A \in_R Z_q^{n \times m}$ and $b = s^t A + e \in Z_q^m$.
- Encryption: To encrypt each bit of message, do the following.

  Choose a string $x \in_R \{0,1\}^m$.

  Compute $u = \sum x_i a_i = Ax$, $u' = bit \left\lfloor q/2 \right\rfloor + b^t x$.
- Decryption: Compute $u' - s^t u = bit \left\lfloor q/2 \right\rfloor + ex$.

  Output is 0 if $u' - s^t u$ is closer to 0 than $\left\lfloor q/2 \right\rfloor$ and 1 otherwise.

Dual Public Key Cryptosystem [GPV08]

- System Parameter: Integers n (the security parameter), m (number of equations), q modulus, and a real α > 0 (noise parameter).
- Private Key: is a vector $x \in_R \{0,1\}^m$.
- Public Key: is $A \in_R Z_q^{n \times m}$ and $b = Ax$.

Encryption Matrix

- Encryption: To encrypt each bit of message, do the following.

  Choose a $s \in_R Z_q^n$.

  Compute $u = s^t A + e$, $u' = bit \left\lfloor q/2 \right\rfloor + s^t b + e'$.

- Decryption: Compute $u' - ux = bit \left\lfloor q/2 \right\rfloor + e' - ex$.

  Output is 0 if $u' - ux$ is closer to 0 than $\left\lfloor q/2 \right\rfloor$ and 1 otherwise.

Correctness:

- Error = $\sum x_i a_i = ex$. Error is atmost m normal error terms each with standard deviation αq and mean zero.

  Sum of normal distribution is also normal distribution with mean zero and variance $(\sigma^2) = m\alpha^2 q^2 \le \frac{q^2}{(\log n)^3}$, since $\alpha = \frac{1}{\sqrt{n}(\log n)^2}$.

  $$q/4 = \frac{(\log n)^{3/2}}{4} \times \frac{q}{(\log n)^{3/2}} \ge 10\sigma$$

- Hence probability that error term is greater than q/4 is negligible.

LWE based IBE [GPV 08]

Random Oracle $H:\{0,1\}^* \to Z_q^n$ that maps identities to public key of the dual cryptosystem. $u = H(id)$: public key.

- IBE Setup: Generate a trapdoor function f_A with trapdoor T. The master public key is A, master secret key is T (Algo TrapGen(q,n)).

$$f_A(e) = Ae \bmod q$$

- IBE Extract (A,T,id): Let $u = H(id)$ and choose a decryption key $e \leftarrow f_A^{-1}(u)$ using preimage sampler with trapdoor T. Store (id,e) and return e.
- Encryption: Dual Encryption.
- Decryption: Dual Decryption.

Lattice IBE in the Standard Model for selective ID (ABB10)

- Master Secret Key : short basis for A_0 .
- Two uniformly random $n\times m$ matrices A_1 and B in $Z_q^{n\times m}$ and a uniformly random n-vector $u \in Z_q^n$. $H:\{0,1\}^* \to Z_q^{n\times m}$.
- Encryption Matrix $F_{id} = [A_0 | A_1 + H(id)B]$
- Secret key for Id is short vector x such that $F_{id}x = u$.
- Sample left Algorithm (A_0, M_1, T, u):

  Let $F_{id} = [A_0 | M_1]$, The algorithm outputs x such that $F_{id}x = u$.

Simulation: Challenger Identity = Id*.

- Challenger does not have basis for A_0 but have basis for B.
- Choose $A_1 = A_0R - H(id^*)B$, where R is low norm.
- $F_{id} = [A_0 | A_0R + (H(id) - H(id^*))B]$
- Sample Right Algorithm (A_0, B, R, T_B, u): Let $F_{id} = [A_0 | A_0R + B]$. The algorithm outputs short vector x such that $F_{id}x = u$.

Adaptively Secure Lattice IBE in the Standard Model (ABB10)

- Waters showed how to convert the selectively secure IBE to an adaptively secure IBE.
- Using Waters technique,
- $$F_{id} = \left[A_0 \middle| B + \sum_{i=1}^{l} b_i A_i \right]$$

  where $id = (b_1, \ldots, b_l)$ in $\{\mathbf{1}, -\mathbf{1}\}^l$

Our Adaptively Secure Lattice IBE in the Standard Model with short Public Parameter

- ABB[10] requires l $n \times m$ matrices.
- Independent work by Chaterjee and Sarkar[05] and Naccache provided a variant of Waters IBE to reduce public parameters.
- The Idea is to divide an l-bit identity into l' block of l/l' so that size of public parameters can be reduced from l $n \times m$ to l' $n \times m$ matrices.
- Identity $id = (b_1, \ldots, b_l)$ where each b_i an l/l' = β bit string.
- Encryption Matrix: $F_{id} = \left[A_0 \middle| B + \sum_{i=1}^{l'} b_i A_i \right]$
- $$R_{id} = \sum_{i=1}^{l'} b_i R_i \in \left\{ -l'(2^{\beta} - 1), \ldots, l'(2^{\beta} - 1) \right\}$$

  where $R_i \in_R \{-1, 1\}^{m \times m}$

- To satisfy some requirements like error term should be less than q/4 etc.
- New $q = q\left(2^{\beta}\frac{l'}{l}\right)^2 = q\left(\frac{2^{\beta}}{\beta}\right)^2$.
- When public parameters are reduced by factor β the value of q is increased by $\left(\frac{2^{\beta}}{\beta}\right)^2$ or number of bits in q is increased by $(\beta - \log(\beta))^2$.
- Cost of key generation, encryption and decryption is same as ABB[10].
- In our scheme computational cost increases because of increase in value of q.

Space-Time Trade-off

- Relative decrease in amount of space $= \frac{l - l'}{l}$.
- Relative increase in time $= \frac{Z_{q'} - Z_q}{Z_q} = \frac{(\beta - \log \beta)^2}{|Z_q|}$.
- For $l = 160$ and $|Z_q| = 512$.

l'	Relative decrease in space	Relative increase in time
8	95	48
16	90	8.71
32	80	1.40
64	60	0.27

Our Adaptively Secure Lattice Hierachical IBE in the Standard Model with short Public Parameters

- ABB[10] constructed selective -ID secure lattice HIBE.
- Using Waters idea, we convert selective-ID secure lattice HIBE to adaptively secure lattice HIBE.
- Then using blocking technique we reduce public parameters.

Setup(l,λ): TrapGen(q,n) generate a matrix A_0 and a short basis T.

- Select $l''l+1$ uniformly random $n \times m$ matrices $A_{1,1},\ldots,A_{l,l''}$ and B.
- Select a uniformly random n-vector u.

Derive: Encryption matrix $$F_{id/id_l} = \left[A_0 \middle| \sum_{i=1}^{l''} A_{1,i} b_{1,i} + B \middle| \ldots \middle| \sum_{i=1}^{l''} A_{l,i} b_{l,i} + B \right]$$

or $$F_{id/id_l} = \left[F_{id/id_{l-1}} \middle| \sum_{i=1}^{l''} A_{l,i} b_{l,i} + B \right]$$ where $id/id_l = (id_1,\ldots,id_l)$

and $id_i = (b_{i,1},\ldots,b_{i,l''})$

Using SampleLeftAlgorithm, secret key for ID is answered.

Encryption:

$$C_0 = s^T u_0 + x + bit\lfloor q/2 \rfloor$$

$$C_1 = s^T F_{id} + \begin{bmatrix} y \\ \hline z \end{bmatrix}$$

Decryption:

$$C_0 - e_{id}^T C_1 \approx bit$$

Security

Simulation:

- Challenger does not have short basis for A_0 but have short basis for B.
- Choose $A_{k,i} = A_0 R_{k,i} + h_{k,i} B$.
- Encryption matrix for $id / id_l = (id_1, \ldots, id_l)$ is

$$F_{id/id_l} = \left[A_0 \middle| \sum_{i=1}^{l''} A_{1,i} b_{1,i} + B \middle| \ldots \middle| \sum_{i=1}^{l''} A_{l,i} b_{l,i} + B \right]$$

Substituting the value of matrix $A_{k,i}$

$$F_{id/id_l} = \left[A_0 \middle| A_0 R_{id} + B h_{id} \right]$$

where

$$R_{id} = \sum_{i=1}^{l''} R_{1,i} b_{1,i} \middle| \ldots \middle| \sum_{i=1}^{l''} R_{l,i} b_{l,i}$$

and

$$h_{id} = (1 + \sum_{i=1}^{l''} h_{1,i} b_{1,i}) \middle| \ldots \middle| (1 + \sum_{i=1}^{l''} h_{l,i} b_{l,i})$$

Abort Resistant Hash Function

Definition: Let $H = \{\hbar : X \to Y\}$ be family of hash functions from X to Y where $0 \in Y$. For a set of Q+1 inputs $\bar{x} = (x_0, \ldots, x_Q) \in x^{Q+1}$, define the non-abort probability of $\bar{x}$ as the quantity.

$$\alpha(\bar{x}) = \Pr[\hbar(x_0) = 0 \wedge \hbar(x_1) \ldots \wedge \hbar(x_Q) \neq 0]$$

$$\hbar_{id} = \hbar(id_1, \ldots, id_l) = (1 + \sum_{i=1}^{l"} h_{1,i} b_{1,i}) | \ldots | (1 + \sum_{i=1}^{l"} h_{l,i} b_{l,i})$$

Lemma: Let q be prime and 0<Q<q. Then the hash function family defined is $\left(Q, \frac{1}{q}\left(1 - \frac{Q}{q^l}\right), \frac{1}{q^l}\right)$ resistant.

Theorem for the Security of Our Scheme: Suppose there exixts a probabilistic algorithm A (Adversary) that wins the IND-ID-CPA game with advantage ε, making no more than $Q \geq q^l / 2$ adaptive chosen queries. Then there is a probabilistic algorithm B that solves LWE problem in about the same time as A and with $\varepsilon' \geq \varepsilon / 4q^l$.

Thank You!