Section 5

Linearity

LWE Symmetric Encryption


```
Gen():
   s ← Z_q^n
   return s

Enc(s,m):
   a ← Z_q^n
   e ← χ
   b = ⟨a,s⟩ + e + (q/p)m
   return (a,b)

Dec(s,(a,b)):
   d = b - ⟨a,s⟩ mod q
   return (round(d*p/q))
```

Compact (Matrix) LWE


```
Gen():
   S ← Z_q^{l×n}
   return S

Enc(S,M) = (A,B)
   A ← Z_q^{n×w}
   E ← χ^{l×w}
   B = SA + E + round((p/q)M) mod q

Dec(S,(A,B)):
   D ← B - SA mod q
   return round(D*p/q)
```

Notation:

- $[A, B]$: horizontal concatenation
- (A, B): vertical concatenation

Linearity of the LWE function

- Let `LWE(S,X;A,E)= SA + X + E` be the *raw* LWE function
- Encryption: `Enc(S,M)= LWE(S,(q/p)M; A,E)` for random `A,E`
- Linear properties:

```
LWE(S,X;A,E) + LWE(S,X';A',E')
        = LWE(S,X+X';A+A',E+E')

LWE(S,X;A,E) - LWE(S,X';A',E')
        = LWE(S,X-X';A-A',E-E')

c*LWE(S,X;A,E) = LWE(S,c*X; c*A,c*E)
```

Linearity of the LWE function

- Let LWE(S,X;A,E)= SA + X + E be the *raw* LWE function
- Encryption: Enc(S,M)= LWE(S,(q/p)M; A,E) for random A,E
- Linear properties:

```
LWE(S,X;A,E) + LWE(S,X';A',E')
        = LWE(S,X+X';A+A',E+E')

LWE(S,X;A,E) - LWE(S,X';A',E')
        = LWE(S,X-X';A-A',E-E')

c*LWE(S,X;A,E) = LWE(S,c*X; c*A,c*E)
```

- Key Homomorphism:

```
LWE(S,X;A,E) + LWE(S',X';A,E')
      = LWE(S+S',X+X'; A, E+E')
```

- Ciphertexts must use the same A!

Linearity of Ciphertexts

Ciphertexts that "encrypt" X under S with error E.

Definition

LWE(S,X;E) = { (A,B): B = LWE(S,X;A,E)}

LWE(S,X;β) = { (A,B) : B = LWE(S,X;A,E), $|E|_\infty < \beta$ }

Linearity of Ciphertexts

Ciphertexts that "encrypt" X under S with error E.

Definition

LWE(S,X;E) = { (A,B): B = LWE(S,X;A,E)}

LWE(S,X;β) = { (A,B) : B = LWE(S,X;A,E), $|E|_\infty < \beta$ }

- LWE(S,X;E)+ LWE(S,X';E')$\subseteq$ LWE(S,X+X';E+E')
- LWE(S,X;E)- LWE(S,X';E')$\subseteq$ LWE(S,X-X';E-E')
- c*LWE(S,X;E)$\subseteq$ LWE(S,c*X; c*E)

Linearity of Ciphertexts

Ciphertexts that "encrypt" X under S with error E.

Definition

LWE(S,X;E) = { (A,B): B = LWE(S,X;A,E)}

LWE(S,X;β) = { (A,B) : B = LWE(S,X;A,E), $|E|_\infty < \beta$ }

- LWE(S,X;E)+ LWE(S,X';E')$\subseteq$ LWE(S,X+X';E+E')
- LWE(S,X;E)- LWE(S,X';E')$\subseteq$ LWE(S,X-X';E-E')
- c*LWE(S,X;E)$\subseteq$ LWE(S,c*X; c*E)

Question

LWE(S,X;β) + LWE(S,X';β') $\subseteq$LWE(S,X+X';$\beta+\beta'$) ?

Question

LWE(S,X;β) - LWE(S,X';β') $\subseteq$LWE(S,X+X';$\beta-\beta'$) ?

Message and Ciphertext Operations

- Addition:
  - $M_0 + M_1 \in \mathbb{Z}_q^{l \times w}$
  - $(A_0, B_0) + (A_1, B_1) = (A_0 + A_1, B_0 + B_1) \in \mathbb{Z}_q^{(n+l) \times w}$
- Subtraction
  - $M_0 - M_1 \in \mathbb{Z}_q^{l \times w}$
  - $(A_0, B_0) - (A_1, B_1) = (A_0 - A_1, B_0 - B_1) \in \mathbb{Z}_q^{(n+l) \times w}$
- Scalar multiplication
  - $c \cdot M \in \mathbb{Z}_q^{l \times w}$
  - $c \cdot (A, B) = (c \cdot A, c \cdot B) \in \mathbb{Z}_q^{(n+l) \times w}$
- Arbitrary linear transformations ...

Additive Homomorphism Encryption

- Homomorphic Encryption supporting the *addition* of ciphertexts

$\mathtt{sk} \leftarrow \mathtt{Gen}()$
$c_0 \leftarrow \mathtt{Enc}(\mathtt{sk}, m_0)$
$c_1 \leftarrow \mathtt{Enc}(\mathtt{sk}, m_1)$
$\mathtt{c} = c_0 + c_1$
$\mathtt{m} = m_0 + m_1$
$\mathtt{Dec}(\mathtt{sk}, \mathtt{c}) \stackrel{?}{=} \mathtt{m}$

Question

Does LWE encryption satisfy the addititive homomorphic property? For what error bound $|\chi| < \beta$?

Question

Is ciphertext c *distributed according to* $\mathtt{Enc}(m_0+m_1)$*?*

Summation

- Homomorphic Encryption supporting the *addition* of ciphertexts

$\mathtt{sk} \leftarrow \mathtt{Gen}()$
$c_1 \leftarrow \mathtt{Enc}(\mathtt{sk}, m_1)$
$c_2 \leftarrow \mathtt{Enc}(\mathtt{sk}, m_2)$
$\ldots$
$c_k \leftarrow \mathtt{Enc}(\mathtt{sk}, m_k)$
$\mathtt{c} = c_1 + c_2 + \ldots + c_k$
$\mathtt{m} = m_1 + m_2 + \ldots + m_k$
$\mathtt{Dec}(\mathtt{sk}, \mathtt{c}) \stackrel{?}{=} \mathtt{m}$

Question

For any given bound $|\chi| < \beta$, what is the largest value of k for which one can add k ciphertexts?

Subtraction and Scalar multiplication

- Subtraction $m_0 - m_1$: similar to addition $m_0 + m_1$
- ± 1-linear combinations: similar to summation
- Scalar multiplication $c \cdot m$: error grows by a factor c
- Ciphertexts can be multiplied only by small scalars!

Concatenation

- `LWE(S,X;A,E)= SA + X + E`
  - $S \in \mathbb{Z}_q^{k \times n}$
  - $A \in \mathbb{Z}_q^{n \times w}$
  - $X, E \in \mathbb{Z}_q^{k \times w}$
- The same `S` can be used with messages `X` with any number of columns w
- Message Concatenation `X | X' = [X,X']`

Definition

`(A,B) | (A',B')= ([A,A'],[B,B'])`

Theorem

`LWE(S,X;A,E)| LWE(S,X';A',E)`$\subseteq$`LWE(S,[X,X'];[A,A'],[E,E'])`

Linear Transforms

- Left multiplication by a constant matrix: `M` $\rightarrow$ `M T`
- Ciphertext `C = LWE(S,M;E)`
- Notice: `M` and `C` have the same number of columns
- We can apply `T` to `C`: `C` $\rightarrow$ `CT`

Theorem

`LWE(S,X;A,E)*T` $\subseteq$ `LWE(S,XT;AT,ET)`

`LWE(S,X;E)*T` $\subseteq$ `LWE(S,XT;ET)`

Special case:

- Addition: `C + C' = [C|C']T` for `T=(I,I)`
- Subtraction: `C - C' = [C|C']T` for `T=(I,-I)`

Constant Messages

Question

Can you compute an LWE encryption of a message `M` *without knowing the secret key* `S`*?*

- I pick `S` $\leftarrow$ `Gen()` and keep it secret
- Goal: find ciphertext `C` such that `Dec(S,C)= M`

Constant Messages

Question

Can you compute an LWE encryption of a message `M` *without knowing the secret key* `S`*?*

- I pick `S` $\leftarrow$ `Gen()` and keep it secret
- Goal: find ciphertext `C` such that `Dec(S,C)= M`
- Let `(A,B) = (0,(q/p)M)`
- `Dec(S,(A,B))= (p/q)(B - SA)= M`
- We write `Const(M)` for the constant ciphertext `(0,(q/p)M)`
- Remarks:
  - The ciphertext `C` is independent of `S`
  - `C = LWE((q/p)M;0)` is a "noiseless" encryption of `M`

Constant Messages as Homomorphic properties

- `LWE(S,M;E)+ LWE((q/p)M';0)= LWE(S,M+M';E)`
- Homomorphism for "nullary functions" $f_M() = M$
  - Given an empty sequence of ciphertexts [], produce an encryption of $f_M([]) = M$
- Homomorphism for unary functions $f_M(M') = M + M'$
  - Given an encryption of M', produce an encryption of the shifted message $M + M'$

Circular security

- A PKE scheme is "circular secure" if one can securly publish the encryption `Enc(pk,sk)`.
- A SKE scheme is "circular secure" if one can securly publish the encryption `Enc(sk,sk)`.

Definition

A PKE scheme `(Gen,Enc,Dec)` is circular secure if `(Gen',Enc',Dec)` is IND-CPA secure where

```
Gen'():
   (sk,pk) ← Gen()
   ct ← Enc(pk,sk)
   pk' = (pk,ct)

Enc'((pk,ct),msg) = Enc(pk,msg)
```

Application: Public key encryption

- Can we transform Secret Key Encryption to Public Key Encryption?
  - Not in general: black box separations
  - Impagliazzo's worlds: Minicrypt vs Cryptomania

Application: Public key encryption

- Can we transform Secret Key Encryption to Public Key Encryption?
  - Not in general: black box separations
  - Impagliazzo's worlds: Minicrypt vs Cryptomania
- What if we start from an Additively Homomorphic SKE scheme?
  - Black box separation results break down
- What about a weakly (bounded) additive scheme?
- What about our LWE SKE scheme?

PKE: Construction

- Start from SKE (Gen,Enc,Dec)
- Construct a PKE (Gen',Enc',Dec)

```
Gen'():
   sk ← Gen()
   for i=1..n
      pk[i] ← Enc(sk,0)
   pk = pk[1..n]
   return (sk,pk)

Enc'(pk,msg):
   for i=1..n
     r[i] ← {0,1}
   ct = Const(msg) + sum { pk[i] : r[i] = 1 }
   return ct
```

Correctness of PKE


```
Dec(sk, msg+Enc(sk,0)+...+Enc(sk,0))
    = msg + 0 + ... + 0 = msg
```

Theorem

If SKE is (1-hop) homomorphic under constant increment and n-summation, then PKE is correct.

Theorem

If SKE is (1-hop) homomorphic under constant increment and hn-summation, then PKE is correct and homomorphic under constant increment and n-summation.

Correctness of PKE


```
Dec(sk, msg+Enc(sk,0)+...+Enc(sk,0))
    = msg + 0 + ... + 0 = msg
```

Theorem

If SKE is (1-hop) homomorphic under constant increment and n-summation, then PKE is correct.

Theorem

If SKE is (1-hop) homomorphic under constant increment and hn-summation, then PKE is correct and homomorphic under constant increment and n-summation.

Question

Assume SKE is an IND-CPA secure and homomorphic. Is PKE secure?

- For what value of n?
- Certainly not secure for $n = 1$ (or even $n = 0$!)
- What about large n?
- How large?
- Answer: Secure, for large enough n and any additively homomorphic SKE [Rothblum, TCC 2011]

The case of LWE SKE

- Consider the PKE scheme obtained from our LWE-based SKE

```
Gen'():
   S ← Gen()
   P = Enc(S,0)|..|Enc(S,0) = Enc(S,[0..0])
   return (S,P)

Enc'(P,M):
   R ← {0,1}*
   PR + Const(M)
```

Theorem

LWE PKE is RR-IND secure.

Universal Hashing

Definition

A function family $H = \{h : X \to Y | h\}$ is 2-universal if for any $a, b \in X$,

$$\{(h(a), h(b)) | h \in H\} \equiv \{(f(a), f(b)) | f : X \to Y\}$$

- Let $(X, +)$ be an additive group
- For any vector $a \in X^n$, define the subset-sum function $h(a, S) = \sum\{a_i : i \in S\}$

Question

Which of the following function families is 2-universal?

1. $\{h_a : S \to h(a, S) | a \in X^n\}$
2. $\{h_S : a \to h(a, S) | S \subseteq \{1, .., n\}\}$
3. *Both*
4. *Neither*

Universal Hashing (continued)

- $h_a(S) = \sum_{i \in S} a_i$ is not 2-universal
- What about $g_{a,b}(S) = b + h_a(S)$?
  - Yes, this is 2-universal
  - Prove it as an exercise
- $\{h_a : \{0,1\}^n \to X\}_a$ still satisfies a weaker property which is enough for our purposes

Definition

For any $a \neq b$, $\Pr_h\{h(a) = h(b)\} = 1/|X|$

- We will refer to this weaker property as 2-universal'

Universal Hashing: proof

Lemma

For any group $(X, +)$, the function family $\{h_a(S) = \sum_{i \in S} a_i\}_a$ is 2-universal', i.e., for all $S \neq T$ we have $\Pr_h\{h(S) = h(T)\} = 1/|X|$

Proof.

- Let $j \in S \setminus T$
- Fix a_i for all $i \neq j$
- Let $T' = T \setminus S$ and $S' = S \setminus (T \cup \{i\})$
- $c = \sum_{i \in T'} a_i - \sum_{i \in S'} a_i$ does not depend on a_j
- $h_a(S) = h_a(T)$ iff $a_j = c$
- $\Pr\{a_j = c\} = 1/|X|$

Leftover Hash Lemma

Lemma

For any 2-universal' family $\{h : X \to Y | h \in H\}$, *the distributions*

- $\{(h, h(x)) | h \leftarrow H, x \leftarrow X\}$
- $\{(h, y) | h \leftarrow H, y \leftarrow Y\}$

are within statistical distance $\Delta \leq \sqrt{|Y|/|X|}$.

Proof Steps:

1. If H is 2-universal', then $(H, H(X))$ has small collision probability
2. If $(H, H(X))$ has small collision probability, then it is statistically close to uniform

Collision Probability and Uniformity

- Z, Z' i.i.d., with $\Pr\{Z = z\} = p(z)$

Definition

Collision Probability:
$C(Z) = \Pr\{Z = Z'\} = \sum_z p(z)^2$

- $\sum_z (p(z) - 1/|Z|)^2 = C(Z) - 1/|Z|$
- Norm inequality: $\forall v \in R^n . \|v\|_1 \leq \sqrt{n}\|v\|_2$
- $\Delta(Z, U) = \frac{1}{2} \sum_z |p(z) - 1/|Z||$

Collision Probability and Uniformity

- Z, Z' i.i.d., with $\Pr\{Z = z\} = p(z)$

Definition

Collision Probability:
$C(Z) = \Pr\{Z = Z'\} = \sum_z p(z)^2$

- $\sum_z (p(z) - 1/|Z|)^2 = C(Z) - 1/|Z|$
- Norm inequality: $\forall v \in R^n . \|v\|_1 \leq \sqrt{n}\|v\|_2$
- $\Delta(Z, U) = \frac{1}{2}\sum_z |p(z) - 1/|Z||$

- $\Delta \leq \frac{1}{2}\sqrt{|Z|}\sqrt{\sum_z (p(z) - 1/|Z|)^2}$
- $\Delta \leq \frac{1}{2}\sqrt{|Z| C(Z) - 1}$

Collision Probability of Universal Hashing

- $Z = (H, H(X))$, 2-universal function family $H : X \to Y$
- Collision Probability of Z:

  $$C(Z) = \Pr(h = h', h(x) = h'(x') | h, h' \leftarrow H, x, x' \leftarrow X)$$

- $C = \frac{1}{|H|} \Pr_{x,x'}[\Pr_h(h(x) = h(x'))]$

Collision Probability of Universal Hashing

- $Z = (H, H(X))$, 2-universal function family $H : X \to Y$
- Collision Probability of Z:

  $$C(Z) = \Pr(h = h', h(x) = h'(x') | h, h' \leftarrow H, x, x' \leftarrow X)$$

- $C = \frac{1}{|H|} \Pr_{x,x'}[\Pr_h(h(x) = h(x'))]$
- Union bound:
  - $\Pr(x = x') = 1/|X|$
  - If $x \neq x'$, then $\Pr_h(h(x) = h(x')) \leq 1/|Y|$

Collision Probability of Universal Hashing

- $Z = (H, H(X))$, 2-universal function family $H : X \to Y$
- Collision Probability of Z:

  $$C(Z) = \Pr(h = h', h(x) = h'(x') | h, h' \leftarrow H, x, x' \leftarrow X)$$

- $C = \frac{1}{|H|} \Pr_{x,x'}[\Pr_h(h(x) = h(x'))]$
- Union bound:
  - $\Pr(x = x') = 1/|X|$
  - If $x \neq x'$, then $\Pr_h(h(x) = h(x')) \leq 1/|Y|$
- $C \leq \frac{1}{H}(\frac{1}{|X|} + \frac{1}{|Y|})$

Collision Probability of Universal Hashing

- $Z = (H, H(X))$, 2-universal function family $H : X \to Y$
- Collision Probability of Z:

  $$C(Z) = \Pr(h = h', h(x) = h'(x') | h, h' \leftarrow H, x, x' \leftarrow X)$$

- $C = \frac{1}{|H|} \Pr_{x,x'}[\Pr_h(h(x) = h(x'))]$
- Union bound:
  - $\Pr(x = x') = 1/|X|$
  - If $x \neq x'$, then $\Pr_h(h(x) = h(x')) \leq 1/|Y|$
- $C \leq \frac{1}{H}(\frac{1}{|X|} + \frac{1}{|Y|})$
- Using $|Z| = |H| \cdot |Y|$ we get

$$\Delta \leq \frac{1}{2}\sqrt{|Z|C - 1} = \frac{1}{2}\sqrt{|Y|/|X|}$$

Security of LWE PKE


```
Gen(): S,E ← ...
       P = Enc(S,[0..0]) = (A,SA+E)
       return (S,P)

Enc(P,M): R ← {0,1}*
          return PR + Const(M)
```

Theorem

LWE PKE is RR-IND secure.

Security of LWE PKE


```
Gen(): S,E ← ...
       P = Enc(S,[0..0]) = (A,SA+E)
       return (S,P)

Enc(P,M): R ← {0,1}*
          return PR + Const(M)
```

Theorem

LWE PKE is RR-IND secure.

Proof:

1. Assume Adv breaks PKE
2. LWE Assumption: P = (A,SA+E)≈(A,B)
3. Adv breaks RR-CPA when P is uniform
4. If P is uniform, then (P,PR) is close to uniform

Details

Claim: $(\mathtt{P}, \mathtt{PR})$ is close to uniform

- Enough to look at a single column $(\mathtt{P}, \mathtt{Pr})$
  - Statement for matrix $(\mathtt{P}, \mathtt{PR})$ follows by hybrid argument
- $\mathtt{P}: \mathtt{r} \to \mathtt{Pr}$ is 2-universal
  - Columns of $\mathtt{P}$ belong to a group $(\mathbb{Z}_q^{n+l}, +)$
  - $\mathtt{r}$ selects a subset of the columns of $\mathtt{P}$
  - Apply Leftover Hash Lemma

Homomorphic PKE

- Enc(P,M)= PR + Const(M)
- Enc(P,M)+ Enc(P,M')= PR + Const(M)+ PR' + Const(M')= P(R+R')+ Const(M+M')
- Enc(P,M)+ Enc(P,M')≈Enc(P,M+M')
  - Noise: E+E'

Homomorphic PKE

- Enc(P,M)= PR + Const(M)
- Enc(P,M)+ Enc(P,M')= PR + Const(M)+ PR' + Const(M')= P(R+R')+ Const(M+M')
- Enc(P,M)+ Enc(P,M')≈Enc(P,M+M')
  - Noise: E+E'
- [Enc(P,M)| Enc(P,M')] = Enc(P,[M|M'])
  - Noise: [E|E']
- Enc(P,M)T ≈Enc(P,MT)
  - Noise: ET
  - T must be small

Encoding modulo q

- Ciphertext modulus q. Assume $q = 2^k$
- Plaintest modulus $p \ll q$, e.g., p=2. Use scaling `Const(msg)= (0,(q/p)msg)` to allow error correction and correct decryption

Encoding modulo q

- Ciphertext modulus q. Assume $q = 2^k$
- Plaintest modulus $p \ll q$, e.g., p=2. Use scaling `Const(msg)= (0,(q/p)msg)` to allow error correction and correct decryption
- What if we want to encrypt `msg` $\in \mathbb{Z}_q$?

Encoding modulo q

- Ciphertext modulus q. Assume $q = 2^k$
- Plaintest modulus $p \ll q$, e.g., p=2. Use scaling Const(msg)= (0, (q/p)msg) to allow error correction and correct decryption
- What if we want to encrypt msg $\in \mathbb{Z}_q$?
- Idea:
  - write $msg = \sum_i m_i 2^i$, where $m_i \in \{0, 1\}$
  - Encrypt each bit individually: $\mathsf{Enc}(m_0), \ldots, \mathsf{Enc}(m_k)$

Encoding modulo q

- Ciphertext modulus q. Assume $q = 2^k$
- Plaintest modulus $p \ll q$, e.g., p=2. Use scaling Const(msg)= (0,(q/p)msg) to allow error correction and correct decryption

```
Enc(m: {0,1}^k) = (a, Sa+e+(q/2)m)

bitDecomp(msg: Z_q) =
   for i=0..k-1
     m[i] = (msg >> i) mod 2
   return m[]

Enc'(msg: Z_q) =
   return (Enc(bitDecomp(msg))
```

Linear Encoding

- Bit encoding: $(\mathtt{msg}:\mathbb{Z}_q) \to (\mathtt{m[*]}:\{0,1\}^k)$
  - good: works for any message space
  - bad: breaks linear homomorphic properties
- We need to use a linear encoding function:
  - $(\mathtt{msg}:\mathbb{Z}_q) \to (\mathtt{m[*]}:\mathbb{Z}_q^k)$
  - $\mathtt{msg} \to \mathtt{msg}*(1,2,4,8,\ldots)$

Linear Encoding

- Bit encoding: $(\texttt{msg}:\mathbb{Z}_q) \to (\texttt{m[*]}:\{0,1\}^k)$
  - good: works for any message space
  - bad: breaks linear homomorphic properties
- We need to use a linear encoding function:
  - $(\texttt{msg}:\mathbb{Z}_q) \to (\texttt{m[*]}:\mathbb{Z}_q^k)$
  - $\texttt{msg} \to \texttt{msg*(1,2,4,8,...)}$
- Column encoding:
  - `pow2col = (1,2,4,8,...)`
  - `Enc'(S,msg)= LWE(S,msg*pow2col)= (a,b)`
- Row encoding:
  - `pow2row = [1,2,4,8,...]`
  - `Enc'(s,msg)= LWE(s,msg*pow2row)= (A,b)`

Decoding modulo q

Question

- *Can you decrypt*

  `Enc'(S,msg)= LWE(S,msg*pow2col)= (a,b)`*?*
- *Can you decrypt*

  `Enc'(s,msg)= LWE(s,msg*pow2row)= (A,b)`*?*
- *For what error bound* $|e|_\infty < \beta$*?*

Decryption algorithm

- `Enc'(s,msg)= LWE(s,msg*pow2row)= (A,b)` where
  `b = sA+e+msg*pow2row`

```
Dec'(s,(A,b)):
  msg ← 0
  for i=0...(k-1)
    ct ← (A[k-i-1],b[k-i-1] - msg*2^(k-i))
    m[i] ← Dec(s,ct)
    msg ← msg+m[i]<<(i)
  return msg
```

Decryption algorithm

- `Enc'(s,msg)= LWE(s,msg*pow2row)= (A,b)` where
  `b = sA+e+msg*pow2row`

```
Dec'(s,(A,b)):
  msg ← 0
  for i=0...(k-1)
    ct ← (A[k-i-1],b[k-i-1] - msg*2^(k-i))
    m[i] ← Dec(s,ct)
    msg ← msg+m[i]<<(i)
  return msg
```

Theorem

`(Gen,Enc',Dec')` *is a valid encryption algorithm for* $\beta = q/4$

Question

Does a similar algorithm work for `pow2col`*?*

Arbitrary linear transformations

- Starting point: Enc() linearly homomorphic for small t
  - Enc(P,m)*t ≈Enc(P,mt)
  - problem: error grows by a factor t

Arbitrary linear transformations

- Starting point: `Enc()` linearly homomorphic for small `t`
  - `Enc(P,m)*t` $\approx$ `Enc(P,mt)`
  - problem: error grows by a factor `t`
- What about computations modulo q?
  - **`pow2row`** `= [1,2,4,8,...]`
  - `Enc'(s,msg)= LWE(s,msg*pow2row)= (A,b)`
- Multiplying by any $t \in \mathbb{Z}_q$
  - Compute `tBin[]` = **`bitDecomp`**`(t)`
  - Compute scalar product with vector `tBin[]`

Correctness of scalar multiplication


```
Enc'(s,msg) * tBin[]
    = LWE(s,msg*pow2row;e) * tBin[]
    = LWE(s,msg*pow2row*tBin[];e*tBin[])
    = LWE(s,msg*t;e')
```

- `pow2row * tBin[] =` $\sum_i 2^i \cdot$`tBin[i] = t`

Correctness of scalar multiplication


```
Enc'(s,msg) * tBin[]
    = LWE(s,msg*pow2row;e) * tBin[]
    = LWE(s,msg*pow2row*tBin[];e*tBin[])
    = LWE(s,msg*t;e')
```

- `pow2row * tBin[] =` $\sum_i 2^i \cdot$`tBin[i] = t`
- if $|e| < \beta$, then $|e'| = |\sum_i e_i \cdot tBin[i]| \leq k \cdot \beta$
- Error grows only by $k = \log q$

Correctness of scalar multiplication


```
Enc'(s,msg) * tBin[]
    = LWE(s,msg*pow2row;e) * tBin[]
    = LWE(s,msg*pow2row*tBin[];e*tBin[])
    = LWE(s,msg*t;e')
```

- `pow2row * tBin[] =` $\sum_i 2^i \cdot$ `tBin[i] = t`
- if $|e| < \beta$, then $|e'| = |\sum_i e_i \cdot tBin[i]| \leq k \cdot \beta$
- Error grows only by $k = \log q$
- Problem:
  - result `msg*t` is a value modulo q
  - `Enc(s,msg*t;e')` is not properly encoded
  - we need an encryption of `msg*t*pow2row`

Constant Multiplication algorithm

- `Enc'(s,msg)= LWE(s,msg*pow2row)`
- `Enc'(s,msg)* bitDecomp(t)= LWE(s,msg*t;e')`

```
CMul(C,t):
   T = bitDecomp(t * pow2row)
   return C * T
```

Proof:

Extensions and Generalizations

- Matrix messages

  $M \otimes$ **pow2row** = [M, M*2, M*4, M*8, ...]

- Arbitrary message modulus:

  round(m*(q/p), m*(q/p)/2, m*(q/p)*4, ...)

- Other gadgets, e.g., based on Chinese Remainder Theorem
  - $q = \prod_i p_i$ product of small primes
  - encoding vector **crtRow** = [q/p_1, q/p_2, ..., q/p_k]
  - **crtRow** * crtDecomp(t)= t

Summary

At this point we have an encryption algorithm

$$\texttt{Enc'(S,M)} = \texttt{LWE(S,M} \otimes \texttt{pow2row)}$$

with message space $\mathbb{Z}_q^{w \times l}$, and supporting the homomorphic evaluation of the following operations:

- `Const(M)`: noiseless encryption of `M`
- `(+)`: addition of ciphertexts
- `(-)`: subtraction of ciphertexts
- `CMul(.,T)`: multiplication by any linear transformation modulo `q`