So far, we have shown how to build symmetric crypto and public-key crypto from standard lattice assumptions (e.g., SIS and LWE)

But it turns out, lattices have much additional structure $\Rightarrow$ enable many new advanced functionalities not known to follow from many other standard assumptions (e.g., discre log, factoring, pairings, etc.)

We will begin by studying *fully homomorphic encryption* (FHE)
$\hookrightarrow$ encryption scheme that supports arbitrary computation on encrypted data [very useful for outsourced computation]

Abstractly: given encryption ct_x of value x under some public key, can we derive from that an encryption of $f(x)$ for an arbitrary function f?
- So far, we have seen examples of encryption schemes that support one type of operation (e.g., addition) on ciphertexts
  - ElGamal encryption (in the exponent): homomorphic with respect to addition
  - Regev encryption: homomorphic with respect to addition
- For FHE, need homomorphism with respect to two operations: addition and multiplication
  Major open problem in cryptography (dates back to late 1970s!) — first solved by Stanford student Craig Gentry in 2009
  $\hookrightarrow$ revolutionized lattice-based cryptography!
  $\hookrightarrow$ very surprising this is possible: encryption needs to "scramble" messages to be secure, but homomorphism requires preserving structure to enable arbitrary computation

General blueprint: 1. Build somewhat homomorphic encryption (SWHE) — encryption scheme that supports bounded number of homomorphic operations
2. Bootstrap SWHE to FHE (essentially a way to "refresh" ciphertext)
Focus will be on building SWHE (has all of the ingredients for realizing FHE)

Starting point: Regev encryption

$$\left.\begin{array}{ll} pk: & A = \begin{bmatrix} \bar{A} \\ \bar{s}^T\bar{A} + e^T \end{bmatrix} \in \mathbb{Z}_q^{n\times m} \\ sk: & s^T = [-\bar{s}^T \mid 1] \in \mathbb{Z}_q^n \end{array}\right\} \text{Invariant: } s^TA = e^T$$

$$ct: \quad r \xleftarrow{R} \{0,1\}^m, \quad c \leftarrow Ar + \begin{bmatrix} 0^{n-1} \\ \lfloor q/2 \rceil \cdot \mu \end{bmatrix}$$

$$\hookrightarrow s^Tc = s^T\left(Ar + \begin{bmatrix} 0^{n-1} \\ \lfloor q/2 \rceil \cdot \mu \end{bmatrix}\right) = e^Tr + \lfloor q/2 \cdot \mu \rceil.$$

as long as e^Tr is small, decryption succeeds

We can easily extend the ciphertext to be a matrix (this provides a redundant encoding of the message μ):
- Pad the matrix $\hat{A} = \begin{bmatrix} A \\ 0^{(m-n)\times m} \end{bmatrix} \in \mathbb{Z}_q^{m\times m}$ and the key $\hat{s} = \begin{bmatrix} s \\ 0^{m-n} \end{bmatrix} \in \mathbb{Z}_q^m$ $\Bigg\}$ $\hat{s}^T\hat{A} = s^TA = e^T$

- To encrypt, sample $R \xleftarrow{R} \{0,1\}^{m\times m}$ and compute

$$C \leftarrow \underbrace{\hat{A}R}_{\begin{bmatrix} AR \\ 0^{(m-n)\times m} \end{bmatrix}} + \mu \cdot \lfloor \tfrac{q}{2} \rceil \cdot \begin{bmatrix} I_n & 0^{n\times(m-n)} \\ 0^{(m-n)\times n} & 0^{(m-n)\times(m-n)} \end{bmatrix}$$

$\leftarrow$ security unaffected (LWE + LHL)

Consider decryption:

$$\hat{s}^TC = \hat{s}^T\hat{A}R + \mu \cdot \lfloor \tfrac{q}{2} \rceil \cdot \hat{s}^T\left[\begin{array}{c|c} I_n & 0 \\ \hline 0 & 0 \end{array}\right]$$

$$= e^T R + \mu \cdot \lceil \tfrac{q}{2} \rfloor \cdot \hat{s}^T$$

$$\approx \mu \cdot \lceil \tfrac{q}{2} \rfloor \cdot \hat{s}^T$$

$\longleftarrow$ Decrypt as usual since $\hat{s}$ contains a component with value 1

<u>Observation</u>: C is a ciphertext and $\hat{s}$ is a left eigenvector of $\hat{C}$ with associated eigenvalue $\mu \cdot \lceil \tfrac{q}{2} \rfloor$.

Suppose for a moment that this was an exact eigenvalue (and we do <u>not</u> scale μ).

no scaling needed if there is no error ü

Then, suppose $\hat{s}^T C_1 = \mu_1 \hat{s}^T$ and $\hat{s}^T C = \mu_2 \hat{s}^T$

- Eigenvalues add: $\hat{s}^T (C_1 + C_2) = \mu_1 \hat{s}^T + \mu_2 \hat{s}^T = (\mu_1 + \mu_2) \hat{s}^T$
- Eigenvalues multiply: $\hat{s}^T C_1 C_2 = \mu_1 \hat{s}^T C_2 = \mu_1 \mu_2 \hat{s}^T$

} fully homomorphic!

What about the error?

Back to Regev:
$$\hat{s}^T C_1 = e^T R_1 + \mu_1 \cdot \lceil \tfrac{q}{2} \rfloor \cdot \hat{s}^T$$
$$\hat{s}^T C_2 = e^T R_2 + \mu_2 \cdot \lceil \tfrac{q}{2} \rfloor \cdot \hat{s}^T$$

<u>Addition</u>: $\hat{s}^T (C_1 + C_2) = e^T (R_1 + R_2) + (\mu_1 + \mu_2) \cdot \lceil \tfrac{q}{2} \rfloor \cdot \hat{s}^T$

basically works ; error grows additively

not surprising: Regev is additively homomorphic

<u>Multiplication</u>:
$$\hat{s}^T C_1 C_2 = (e^T R_1 + \mu_1 \cdot \lceil \tfrac{q}{2} \rfloor s^T) C_2$$
$$= e^T R_1 C_2 + \mu_1 \cdot \lceil \tfrac{q}{2} \rfloor \cdot s^T C_2$$
$$= \boxed{e^T R_1 C_2} + \boxed{\mu_1 \cdot \mu_2 \lceil \tfrac{q}{2} \rfloor^2} + \boxed{\mu_1 \cdot \lceil \tfrac{q}{2} \rfloor \cdot e^T R_2}$$

$e^T R_1$ is small, but C_2 is not! | not the right form... | if $\mu_1 = 1$, also large

$\longleftarrow$ lots of problems!!

<u>Main issue</u>: error term from one ciphertext multiplies with a ciphertext during homomorphic multiplication $\longrightarrow$ noise blows up

<u>Solution</u>: Use the gadget matrix (i.e. bit decomposition) to reduce matrix sizes!

Gentry - Sahai - Waters (GSW) FHE:

- Setup (1^λ): Sample $\bar{A} \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, $\bar{s} \xleftarrow{R} \mathbb{Z}_q^n$, $e \leftarrow \chi^m$ $\longrightarrow$ $pk = A = \begin{bmatrix} \bar{A} \\ \bar{s}^T \bar{A} + e^T \end{bmatrix}$, $sk = s = [-\bar{s} \mid 1]$ $\quad (s^T A = e^T)$
- Encrypt (A, μ): $R \xleftarrow{R} \{0,1\}^{m \times m}$
  $C \leftarrow AR + \mu \cdot G \in \mathbb{Z}_q^{n \times m}$ $\longleftarrow$ new message embedding
- Decrypt (s, C): compute $s^T C G^{-1}(\tfrac{q}{2} \cdot I_n)$ and round as usual

<u>Correctness</u>:
$$s^T C G^{-1}(\tfrac{q}{2} \cdot I_n) = s^T (AR + \mu \cdot G) G^{-1}(\tfrac{q}{2} \cdot I_n)$$
$$= \underbrace{e^T R G^{-1}(\tfrac{q}{2} I_n)} + \tfrac{q}{2} s^T$$

suppose e is B-bounded

$\hookrightarrow \|e^T R G^{-1}(\tfrac{q}{2} I_n)\|_\infty \leq m^2 B$

as long as $m^2 B < \tfrac{q}{4}$, scheme is correct

$\uparrow$ if q is power of two or we choose scaling factor to be a power of two, then multiplying by $G^{-1}(\cdot)$ does not change norm $\longrightarrow$ tighten bound to $mB < \tfrac{q}{4}$

<u>GSW invariant</u>: $C = AR + \mu \cdot G$ for some <u>small</u> R.

Decryption succeeds if $m \cdot B \cdot \|R\|_\infty \leq \tfrac{q}{4}$

$\hookrightarrow$ choose $q > 4mB \cdot \|R\|_\infty$

<u>Security</u>: Identical to Regev.

<u>Homomorphism</u>: Suppose
$$C_1 = AR_1 + \mu_1 G$$
$$C_2 = AR_2 + \mu_2 G$$

Addition: $C_1 + C_2$ is encryption of $\mu_1 + \mu_2$:

$$C_1 + C_2 = A(R_1 + R_2) + (\mu_1 + \mu_2) \cdot G$$

New error: $R_+ = R_1 + R_2$, $\|R_+\|_\infty \le \|R_1\|_\infty + \|R_2\|_\infty$

Multiplication: $C_1 G^{-1}(C_2)$ is encryption of $\mu_1 \cdot \mu_2$:

$$\begin{aligned} C_1 G^{-1}(C_2) &= (AR_1 + \mu_1 G) G^{-1}(C_2) \\ &= AR_1 G^{-1}(C_2) + \mu_1 G \cdot G^{-1}(C_2) \\ &= AR_1 G^{-1}(C_2) + \mu_1 C_2 \\ &= AR_1 G^{-1}(C_2) + \mu_1 (AR_2 + \mu_2 G) \\ &= A\underbrace{(R_1 G^{-1}(C_2) + \mu_1 R_2)}_{R_\times} + \mu_1 \mu_2 G \end{aligned}$$

New error: $R_\times = R_1 G^{-1}(C_2) + \mu_1 R_2$, $\|R_\times\|_\infty \le \|R_1\|_\infty \cdot m + \|R_2\|_\infty$

After computing d repeated squarings: noise is $m^{O(d)}$

[for correctness, require that $q > 4mB \cdot \|R\|_\infty$, so bit-length of q scales with multiplicative depth of circuit
↳ also requires super-poly modulus when $d = \omega(1)$
(stronger assumption needed)]

But not quite fully homomorphic encryption: we need a bound on the (multiplicative) depth of the computation

From SWHE to FHE. The above construction requires imposing an a priori bound on the multiplicative depth of the computation. To obtain fully homomorphic encryption, we apply Gentry's brilliant insight of bootstrapping.

High-level idea. Suppose we have SWHE with following properties:

1. We can evaluate functions with multiplicative depth d
2. The decryption function can be implemented by a circuit with multiplicative depth $d' < d$

Then, we can build an FHE scheme as follows:

- Public key of FHE scheme is public key of SWHE scheme and an encryption of the SWHE decryption key under the SWHE public key
- We now describe a ciphertext-refreshing procedure:
  - For each SWHE ciphertext, we can associate a "noise" level that keeps track of how many more homomorphic operations can be performed on the ciphertext (while maintaining correctness).
    ↳ for instance, we can evaluate depth-d circuits on fresh ciphertexts; after evaluating a single multiplication, we can only evaluate circuits of depth-$(d-1)$ and so on ...
  - The refresh procedure takes any valid ciphertext and produces one that supports depth-$(d-d')$ homomorphism; since $d > d'$, this enables unbounded (i.e., arbitrary) computations on ciphertexts

Idea: Suppose $ct_x = \text{Encrypt}(pk, x)$.

Using the SWHE, we can compute $ct_{f(x)} = \text{Encrypt}(pk, f(x))$ for any f with multiplicative depth up to d

Given ct_x, we first compute

$$ct_{ct} = \text{Encrypt}(pk, ct_x)$$ [strictly speaking, encrypt bit by bit]