CS6115: Structure versus Hardness in Cryptography Aug 25, 2021

Lecture 7 : GSW13 Fully Homomorphic Encryption

Lecturer: *Shweta Agrawal* Scribe: *Raghavendra N Vernekar*

1 Introduction

In this course untill now we have studied various developments of Fully Homomorphic Encryption (FHE) schemes. We have seen LWE-based homomorphic encryption schemes. In this lecture we look into the [GSW13] FHE scheme which makes achieving the homomorphic encryption appear much simpler and intuitive.
The key idea used in the schemes discussed so far was multiplication by tensoring. The idea has issues of ciphertext size squaring for each multiplication and also error size squaring up at every step. The technique of *Dimension Reduction* was used for addressing the issue of increase in ciphertext size and the approach of *Bit Decomposition* was looked upon to address the error growth issues. [Bra12] presented the idea of the *Modulus Reduction* technique of scale invariant version to support $\mathcal{O}(n^\epsilon)$ levels. We also have got better understanding of Bootstrapping techniques which transforms the Homomorphic Encryption scheme to Fully Homomorphic Encryption scheme which can support arbitrary polynomial depth circuits. This lecture is dedicated to [GSW13] FHE scheme which is a pretty amazing scheme, conceptually easier to understand and makes achieving the FHE, appear effortless. Lets dive straight into this amazing idea.

2 Gentry, Sahai, Waters 2013 FHE Scheme (GSW13)

We now describe the GSW fully homomorphic encryption scheme[GSW13][MW16].

2.1 Construction

Define a special parity check matrix $\mathbf{G}$. Let $g^t = (1, 2, 4, 8, 16, ..., 2^{k-1} \geq \frac{q}{2})$ where $k = \log q$. Note that the unmentioned entries of $\mathbf{G}$ are all 0.

$$\mathbf{G} = \begin{bmatrix} \cdots g^t \cdots & & & & \\ & \cdots g^t \cdots & & & \\ & & \cdots g^t \cdots & & \\ & & & \ddots & \\ & & & & \cdots g^t \cdots \end{bmatrix} \in \mathbb{Z}_q^{n \times nk}$$

Define an operation $\mathsf{G.BitDec}(x) = x$ where x is a vector of size n. Therefore, define $G^{-1}(C)$ where C is the ciphertext matrix of size $n \times nk$ such that $GG^{-1}(C) = C$. For this to happen $G^{-1}(C)$ should be the Bitdec operation. Hence each entry in M is represented using k bits and hence size of $G^{-1}(C)$ will be $nk \times nk$ and $GG^{-1}(C) = C$. Hence, $G^{-1}(C)$ will be a binary matrix.

Define the GSW FHE scheme as $\mathsf{GSW.FHE} = (\mathsf{GSW.Setup}, \mathsf{GSW.Enc}, \mathsf{GSW.Dec}, \mathsf{GSW.Eval})$.

$\mathsf{GSW.Setup}(1^\lambda)$: On input security parameter λ, it does following:

1. Let $n = n(\lambda, d)$ be the lattice dimension parameters and let $\chi = \chi(\lambda, d)$ be B_χ bounded error distribution. Set modulus q as $q = B_\chi 2^{w(d\lambda \log \lambda)}$ such that $\mathsf{LWE}_{n-1,q,\chi,B_\chi}$ holds.
2. Choose $A' \leftarrow \mathbb{Z}_q^{(n-1)\times m}$, $s' \leftarrow \mathbb{Z}_q^{(n-1)\times 1}$ and sample $e \leftarrow \chi^m$. Here $m = \mathcal{O}(n \log q) = \mathcal{O}(nk)$.
3. Let $b = (s')^t A' + e \in \mathbb{Z}_q^{m\times 1}$
4. We set $s = (-s', 1) \in \mathbb{Z}_q^{n\times 1}$
5. Let $A = \begin{bmatrix} A' \\ b \end{bmatrix} \in \mathbb{Z}_q^{n\times m}$.
6. Output $\mathsf{PK} = A$ and $\mathsf{SK} = s$. (Note that $s^t A = (-s')^t A' + b = -b + b + e = e \approx 0$).

$\mathsf{GSW.Enc}(\mathsf{PK}, \mu)$: On input public key PK and message $\mu \in \{0, 1\}$, it does following:

1. Choose a random matrix as the randomness $R \leftarrow \{0, 1\}^{m\times m}$.
2. Output the encryption of message μ as $C = AR + \mu G \in \mathbb{Z}_q^{n\times m}$.
   (**Note**: We can use m and nk interchangeably since $m = n \log q = nk$. So R matrix can also be seen as $nk \times nk$ matrix.)

$\mathsf{GSW.Dec}(\mathsf{SK}, C)$: On input secret key SK and ciphertext C, it does following:

1. Define $w = (0, 0, 0, \ldots, 0, \lceil \frac{q}{2} \rceil) \in \mathbb{Z}_q^n$.
2. Compute the value of $V = s^t C G^{-1}(w)$.
3. Finally output decrypted message μ' as $| \left\lfloor \frac{V}{q/2} \right\rceil |$
   (**Note**: G^{-1} is just the notation and not the matrix. G^{-1} doesn't even exist as G is not a square matrix. G is the matrix with elements of powers of two.)

$\mathsf{GSW.Eval}(C_1, C_2)$: On input two ciphertexts C_1, C_2, it does following:

1. $\mathsf{GSW.add}(C_1, C_2)$: Output $C_1 + C_2 \in \mathbb{Z}_q^{n\times m}$.
2. $\mathsf{GSW.mult}(C_1, C_2)$: Output $C_\times = C_1 G^{-1}(C_2)$.

Correctness of decryption

Decryption works as follows:

$$\begin{aligned} V &= s^t C G^{-1}(w) \\ &= s^t (AR + \mu G) G^{-1}(w) \\ &= (s^t AR + s^t \mu G) G^{-1}(w) \\ &\approx s^t \mu w \\ &= \mu \left\lceil \frac{q}{2} \right\rceil \end{aligned} \tag{1}$$

We have crucially used the fact that $s^t A \approx 0$ and $G G^{-1}(w) = w$.

Correctness of addition

Addition correctness works as follows: It is evident that addition correctness holds and we will get the value of $\mu_1 + \mu_2$.

$$C_1 + C_2 = AR_1 + \mu_1 G + AR_2 + \mu_2 G$$
$$C_1 + C_2 = A(R_1 + R_2) + (\mu_1 + \mu_2)G$$

Now on decrypting using s^t and obtaining $s^t(C_1 + C_2)$ we obtain

$$s^t(C_1 + C_2) = s^t A(R_1 + R_2) + s^t(\mu_1 + \mu_2)G$$

The first term will be the error term and will be of low norm since $s^t A \approx 0$.

$$s^t(C_1 + C_2) = s^t(\mu_1 + \mu_2)G + error$$
$$s^t(C_1 + C_2) \approx s^t(\mu_1 + \mu_2)G$$

Correctness of multiplication

Multiplication correctness works as follows: Now to obtain the decrypted message we use s^t.

$$\begin{aligned} s^t C_\times &= s^t C_1 G^{-1}(C_2) \\ &= s^t(AR_1 + \mu_1 G)G^{-1}(C_2) \\ &\approx s^t \mu_1 G G^{-1}(C_2) \\ &\approx s^t \mu_1 C_2 \\ &\approx s^t \mu_1 (AR_2 + \mu_2 G) \\ &\approx s^t \mu_1 \mu_2 G \end{aligned} \tag{2}$$

We have kept the error term under the carpet for now and how error (noise) grows is analyzed in the next section.

Theorem 1. *[GSW13] The scheme described above is a secure* FHE *under* $\mathsf{LWE}_{n-1,q,\chi,B_\chi}$ *assumption.*

The proof of security consists of two parts. First we can use the LWE assumption to replace the public key A with a random uniform matrix. Then we can use the Leftover Hash Lemma to replace the ciphertext $C := AR + \mu G$ with a uniformly random value C'. The error growth is analyzed in the next section.

3 Error Growth

In this section, we will analyze the growth of error.

Definition 3.1. (β-noisy ciphertext): A β-noisy ciphertext of some message μ under secret-key $\mathsf{SK} = s \in \mathbb{Z}_q^n$ is a matrix $C \in \mathbb{Z}_q^{n \times m}$ such that: $s^t C = s^t \mu G + e$ for some e with $||e||_\infty \leq \beta$.

Correctness of addition including error

In this addition correctness we will carefully figure out the error value.

Claim 1. *If C_1 is β_1 noisy and C_2 is β_2 noisy, then $C_1 + C_2$ is $(\beta_1 + \beta_2)$ noisy.*

Proof. We know, $C_1 + C_2 = AR_1 + \mu_1 G + AR_2 + \mu_2 G$. Therefore,

$$\begin{aligned} s^t(C_1 + C_2) &= s^t(AR_1 + \mu_1 G + AR_2 + \mu_2 G) \\ &= s^t A(R_1 + R_2) + s^t(\mu_1 + \mu_2)G \\ &= s^t AR_1 + s^t AR_2 + s^t(\mu_1 + \mu_2)G \\ &= e_1 + e_2 + s^t(\mu_1 + \mu_2)G \\ &\leq \beta_1 + \beta_2 + s^t(\mu_1 + \mu_2)G \end{aligned} \tag{3}$$

Hence, if C_1 is β_1 noisy and C_2 is β_2 noisy, then $C_1 + C_2$ is $(\beta_1 + \beta_2)$ noisy. □

Correctness of multiplication including error

In this multiplication correctness we will carefully figure out the error value.

Claim 2. *If C_1 is β_1 noisy and C_2 is β_2 noisy, then error will be $C_\times$ is $(m\beta_1 + \beta_2)$noisy.*

Proof. We know, $C_\times = C_1 G^{-1}(C_2)$. Therefore,

$$\begin{aligned} s^t C_\times &= s^t C_1 G^{-1}(C_2) \\ &= s^t(AR_1 + \mu_1 G)G^{-1}(C_2) \\ &= s^t AR_1 G^{-1}(C_2) + s^t \mu_1 G G^{-1}(C_2) \\ &= e_1 G^{-1}(C_2) + s^t \mu_1 C_2 \\ &= e_1 G^{-1}(C_2) + s^t \mu_1 (AR_2 + \mu_2 G) \\ &= e_1 G^{-1}(C_2) + s^t \mu_1 AR_2 + s^t \mu_1 \mu_2 G \\ &= e_1 G^{-1}(C_2) + e_2 \mu_1 + s^t \mu_1 \mu_2 G \end{aligned} \tag{4}$$

Here β_2 noise is $e_2\mu_1$. μ_1 is a bit and e_2 is low norm, so the error is also low norm. While $||e_1 G^{-1}(C_2)||_\infty$ can be bounded by $\beta_1 m$ as $G^{-1}(C_2)$ is a binary matrix of size $m \times m$ and in worst case, all entries of this matrix can be 1. So, the error generated by multiplication is $m\beta_1 + \beta_2$. □

Correctness of decryption including error

For the correctness of decryption, Let C be the β-noisy encryption of μ. So $s^t C = e + \mu s^t G$ where $||e||_\infty \leq \beta$. Then the decryption equation $V = s^t C G^{-1}(w) = e' + \mu(\frac{q}{2})$ such that $e' = \langle e, G^{-1}(w)\rangle$. So the value of e' is bounded by $m\beta$. The decryption works fine as long as $||e'||_\infty < q/4$. Hence correctness works as long as $\beta < \frac{q}{4m}$.

4 Asymmetry of Noise Growth

According to [BV14], asymmetry allows homomorphic multiplication with additive noise growth.

Let $C_{12} = C_\times$. Let we wish to multiply C_3. According to the way we multiply, error growth changes as described below:

1. Approach 1: Try $C_{123} = C_{12} G^{-1}(C_3)$.

   Here the error term for C_{123} will be $e_{123} = e_{12} G^{-1}(C_3) + \mu_{12} e_3$. Here, e_{12} is polynomial and $G^{-1}(C_3)$ is also polynomial in n. Therefore, we get new error e_{123} as (poly×poly).

2. Approach 2: Try $C_{123} = C_3 G^{-1}(C_{12})$.

   Here the error term for C_{123} will be $e_{123} = e_3 G^{-1}(C_{12}) + \mu_3 e_{12}$. Note that the first term is polynomial $\times$ fresh noise (e_3) and second term is the product of a bit with a polynomial error. So the overall error e_{123} in this case will be (poly + poly) instead of (poly$\times$ poly).

This idea of asymmetry allowing homomorphic multiplication with additive noise growth is discussed in the paper [BV14].

We conclude this lecture having looked at [GSW13] scheme and analyzing its error growth. We will start our ride on some new exciting concepts of Deniable Encryption in the upcoming lectures.

References

[Bra12] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In *Annual Cryptology Conference*, pages 868–886. Springer, 2012.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-based fhe as secure as pke. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 1–12, 2014.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. Cryptology ePrint Archive, Report 2013/340, 2013.

[MW16] Pratyay Mukherjee and Daniel Wichs. Two round multiparty computation via multi-key fhe. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 735–763. Springer, 2016.