CS 4501-6501 Topics in Cryptography	February 11, 2015
Lecture 9	
Lecturer: Mohammad Mahmoody	*Scribe: Natnatee Dokmai*

1 Overview

In this note, we discuss the multiplication and dimension reduction technique of a fully homomorphic encryption scheme from standard LWE assumption based on Regev's public-key cryptosystem [1, 2]. Then, we discuss how to reduce the noise amplified by addition and multiplication using the "bootstrapping" technique introduced by Gentry [3, 4].

2 Multiplication and Dimension Reduction

Recall that we define the encryption of a bit $\mu \in \{0,1\}$ by $\mathsf{Enc}(\mu) = \mathbf{c} = \left(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + e + \mu \left(\frac{q+1}{2}\right)\right) \in \mathbb{Z}_p^{n+1}$ where $\mathbf{a}, \mathbf{s} \in \mathbb{Z}_p^n, b, q \in \mathbb{Z}_p$, and $e \xleftarrow{\$} \chi$ for some noise distribution χ. Let $\mathbf{t} = (-\mathbf{s}, 1) \in \mathbb{Z}_p^{n+1}$, then we can define the decryption scheme by[1]

$$\mathsf{Dec}(\mathbf{c}) = \left\lceil \langle \mathbf{c}, \mathbf{t} \rangle \,/\, \left(\frac{q+1}{2}\right) \right\rfloor = \left\lceil (b - \langle \mathbf{a}, \mathbf{s} \rangle) / \left(\frac{q+1}{2}\right) \right\rfloor = \left\lceil \left(e + \mu \left(\frac{q+1}{2}\right)\right) / \left(\frac{q+1}{2}\right) \right\rfloor = \mu$$

Addition in this scheme can be computed in a straightforward manner by $\mathbf{c}_{\mathsf{add}} \stackrel{\mathsf{def}}{=} c_1 + c_2$. Correctness can be shown as such:

$$\begin{aligned} \mathbf{c}_1 + \mathbf{c}_2 &= \left(\mathbf{a}_1 + \mathbf{a}_2, b = \langle \mathbf{a}_1 + \mathbf{a}_2, \mathbf{s} \rangle + (e_1 + e_2) + (\mu_1 + \mu_2) \left(\frac{q+1}{2}\right)\right) \\ &= \left(\mathbf{a}_{\mathsf{add}}, b_{\mathsf{add}} = \langle \mathbf{a}_{\mathsf{add}}, \mathbf{s} \rangle + e_{\mathsf{add}} + \mu_{\mathsf{add}} \left(\frac{q+1}{2}\right)\right) \end{aligned}$$

Multiplication, however, is more complicated and requires a special multiplication and dimension reduction technique to reduced the blown-up size of ciphertext after multiplication.

2.1 Multiplication

Let $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_p^k$ for some $k \in \mathbb{N}$, we define the *tensor product* of $\mathbf{x}$ and $\mathbf{y}$ by $\mathbf{x} \otimes \mathbf{y} = (x_1 y_1, x_1 y_2, ..., x_n y_n)$ where $\mathbf{x} \otimes \mathbf{y} \in \mathbb{Z}_p^{k^2}$ is the tuple of all the products between every element of $\mathbf{x}$ and $\mathbf{y}$

Using the notations above, we define the multiplication of ciphertext by $\mathbf{c}_{\mathsf{mult}} \stackrel{\mathsf{def}}{=} \mathbf{c}_1 \otimes \mathbf{c}_2$. The

[1] The division and rounding operation here is not in group $\mathbb{Z}_q$. We simply compute $\langle \mathbf{a}, \mathbf{s} \rangle$ in group $\mathbb{Z}_q$ then divide and round it in the real field $\mathbb{R}$, while the output is actually in $\{0, 1\}$.

ciphertext of $\mathbf{c}_{\mathsf{mult}}$ is not decryptable with the secret key $\mathbf{t} \otimes \mathbf{t}$ since

$$\begin{aligned}
2\langle \mathbf{c}_1 \otimes \mathbf{c}_2, \mathbf{t} \otimes \mathbf{t} \rangle &= 2\langle \mathbf{c}_1, \mathbf{t} \rangle \cdot \langle \mathbf{c}_2, \mathbf{t} \rangle \\
&= 2\left(e_1 + \mu_1\left(\frac{q+1}{2}\right)\right)\left(e_2 + \mu_2\left(\frac{q+1}{2}\right)\right) \\
&= (2e_1 + \mu_1)\left(e_2 + \mu_2\left(\frac{q+1}{2}\right)\right) \\
&= (2e_1e_2 + \mu_1e_2 + \mu_2e_1) + \mu_1\mu_2\left(\frac{q+1}{2}\right) \\
&= e_{\mathsf{mult}} + \mu_{\mathsf{mult}}\left(\frac{q+1}{2}\right)
\end{aligned}$$

However, this multiplication technique is not yet acceptable since the dimension of $\mathbf{c}_{\mathsf{mult}}$ as well as $\mathbf{t} \otimes \mathbf{t}$ turns into $\mathbb{Z}_p^{(n+1)^2}$. If we wish to do multiplication polynomially many times, then the dimension of the resulting ciphertext blows up exponentially; this also mean that the party who decrypts the ciphertext also needs to know the computing function, which is an undesirable property for a fully homomorphic encryption scheme. To cope with these issues, we deploy the dimension reduction technique.

2.2 Dimension Reduction

In order to reduce the dimension of resulting ciphertexts from multiplication, we let the key generator generates and publishes the *evaluation* key ek for the evaluator to use to reduce dimension after each multiplication without revealing any information about the secret key.

Let $\mathsf{ek} = \mathbf{D} \in \mathbb{Z}_p^{(n+1)^2 \times (n+1)}$ be a $(n+1)^2 \times (n+1)$ matrix with the property that $\mathbf{D}^T\mathbf{t} = \mathbf{t} \otimes \mathbf{t}$, so that

$$\langle \mathbf{c}, \mathbf{t} \otimes \mathbf{t} \rangle = (\mathbf{t} \otimes \mathbf{t})^T \cdot \mathbf{c} = (\mathbf{D}^T\mathbf{t})^T \cdot \mathbf{c} = \mathbf{t}^T(\mathbf{D} \cdot \mathbf{c}) = \langle \mathbf{D} \cdot \mathbf{c}, \mathbf{t} \rangle$$

Notice that $\mathbf{D} \cdot \mathbf{c} \in \mathbb{Z}_p^{n+1}$ and the decryption key is only $\mathbf{t}$. The next challenge is to find such $\mathbf{D}$ that is both correct and secure.

Let $\tilde{\mathbf{t}} = \mathbf{t} \otimes \mathbf{t}$ and $\tilde{\mathbf{t}}_{(i,j)} = \mathbf{t}[i]\mathbf{t}[j]$ where $i, j \in [n+1]$, i.e. the (i,j)-th element of the tensor product, which is the multiplication of the i-th and j-th element of $\mathbf{t}$. The following $\mathbf{D}$ satisfies our criteria:

$$\mathbf{D}_{(i,j)} = \left(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e_{i,j} + \tilde{\mathbf{t}}_{(i,j)}\right)$$

for $e_{i,j} \xleftarrow{\$} \chi$. (Note that $\mathbf{D}_{(i,j)}$ is the (i,j)-th row of $\mathbf{D}$, the same way we define for $\tilde{\mathbf{t}}$, not the element of $\mathbf{D}$ at row i column j.)

Correctness can be easily shown as follows:

$$\begin{aligned}
\mathbf{D}_{(i,j)}^T\mathbf{t} &= \left(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e_{i,j} + \tilde{\mathbf{t}}_{(i,j)}\right) \cdot (-\mathbf{s}, 1) \\
&= e_{i,j} + \tilde{\mathbf{t}}_{(i,j)} \\
&\approx \tilde{\mathbf{t}}_{(i,j)}
\end{aligned}$$

as long as the error is kept "small". The security follows directly from the standard LWE assumption that

$$\left(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e_{i,j} + \tilde{\mathbf{t}}_{(i,j)}\right) \overset{C}{\approx} (\mathbf{a}, u)$$

where $u \xleftarrow{\$} \mathbb{Z}_p$.

In conclusion, the evaluator can compute one multiplication by $\mathbf{c}'_{\mathsf{mult}} = \mathbf{D} \cdot (\mathbf{c}_1 \otimes \mathbf{c}_2) \in \mathbb{Z}_p^{n+1}$; this decrypts to $\mu_1\mu_2 \left(\frac{q+1}{2}\right) + e_1e_2 + e_{\mathsf{dr}}$ where $e_{\mathsf{dr}} = \sum_i^{n+1} \sum_j^{n+1} c_{i,j}e_{i,j}$.

We can also reduce the amount of noise aggravated by e_{dr} by "interpreting" $\mathbf{c}$ in the binary form, i.e. write $\mathbf{c}$ in $\{0,1\}^{(n+1)^2 \log(q)}$, so that $e_{\mathsf{dr}} = \sum_{i,j \in [(n+1)^2 \log(q)]} e_{i,j}$.

3 Bootstrapping

Addition and multiplication as defined above give us homomorphic encryption, but not yet *fully* homomorphic encryption; the problem stems from accumulative noise as a result of arithmetic operations. For example, addition leaves us with $e_1 + e_2$, and multiplication $e_1e_2 + e_{\mathsf{dr}}$. This is acceptable for a small-sized circuit, i.e. $\mathbf{NC}^1$ circuits, but for a polynomial-sized circuit the noise may grow too large and exceed the *noise bound* (in our case, $[-\frac{p+1}{2}, \frac{p+1}{2}]$), which renders the ciphertext undecryptable. We need a new technique to reduce this noise to make the homomophic encryption scheme fit for any class of circuits to attain fully homomorphic encryption.

Definition 1. ($\mathcal{C}$-homomorphism). *Let $\mathcal{C} = \{\mathcal{C}_\kappa\}_{\kappa \in \mathbb{N}}$ be a class of functions (together with their respective representations). A scheme HE is a $\mathcal{C}$-homomorphic (or, homomorphic for class $\mathcal{C}$) if for any sequence of functions $f_\kappa \in \mathcal{C}_\kappa$ and respective input $\mu_1, ..., \mu_\ell \in \{0,1\}$ (where $\ell = \ell(\kappa)$), it holds that*

$$\Pr\left[\mathsf{HE.Dec}_{\mathsf{sk}}(\mathsf{HE.Eval}_{\mathsf{ek}}(f, c_1, ..., c_\ell)) \neq f(\mu_1, ..., \mu_\ell)\right] = \mathrm{negl}(\kappa),$$

where $(\mathsf{pk}, \mathsf{ek}, \mathsf{sk}) \leftarrow \mathsf{HE.Keygen}(1^\kappa)$ and $c_i \leftarrow \mathsf{HE.Enc}_{\mathsf{pk}}(\mu_i)$.

Definition 2. (compactness). *A homomorphic scheme HE is compact if there exists a polynomial $s = s(\kappa)$ such that the output length of $\mathsf{HE.Eval}(\cdots)$ is at most s bits long (regardless of f or the number of inputs.*

Definition 3. (fully homomorphic encryption). *A scheme HE is fully homomorphic if it is both compact and homomorphic for the class of all arithmetic circuits over $GF(2)$.*

Gentry [3, 4] introduced a noise-reduction technique called *bootstrapping*. The main idea is we build a shallow decryption circuit using the "somewhat" homomorphic encryption scheme that we already have to decrypt a noisy ciphertext to produce a new, fresh ciphertext with lower noise. This decryption circuit includes the decryption key, but does not reveal any information about it since the decryption key is encrypted using somewhat homomorphic encryption and embedded as a part of the circuit. Gentry's bootstrapping theorem is formally stated as follows:

Definition 4. (leveled fully homomorphic encryption). *A leveled fully homomorphic encryption scheme is a homomorphic scheme where the $\mathsf{HE.Keygen}$ gets an additional input 1^L (now $(\mathsf{pk}, \mathsf{ek}, \mathsf{sk}) \leftarrow \mathsf{HE.Keygen}(1^\kappa, 1^L)$) and the resulting scheme is homomorphic for all depth-L binary arithmetic circuits. The bound $s(\kappa)$ on the ciphertext length must remain independent of L.*

Definition 5. (bootstrappable encryption scheme). *Let* HE *be* $\mathcal{C}$*-homomorphic, and let* f_{add} *and* f_{mult} *be the augmented decryption functions of the scheme defined as*

$$f_{\mathsf{add}}^{c_1,c_2} = \mathsf{HE.Dec}_s(c_1) \;\; XOR \;\; \mathsf{HE.Dec}_s(c_2) \quad and \quad f_{\mathsf{mult}}^{c_1,c_2} = \mathsf{HE.Dec}_s(c_1) \;\; AND \;\; \mathsf{HE.Dec}_s(c_2)$$

Then $\mathcal{E}$ *is bootstrappable if*

$$\{f_{\mathsf{add}}^{c_1,c_2}, f_{\mathsf{mult}}^{c_1,c_2} \subseteq \mathcal{C}\}.$$

Namely, the scheme is can homomorphically evaluate f_{add} *and* f_{mult}*.*

Theorem 6. ([3, 4]). *Let* HE *be a bootstrappable scheme, then there exists a leveled fully homomorphic encryption scheme as per Definition 4.*

Definition 7. (weak circular security). *A public key encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is weakly circular secure if it is IND-CPA secure even for an adversary with auxiliary information containing encryptions of all secret key bits:* $\{\mathsf{Enc}_{\mathsf{pk}}(\mathsf{sk}[i])\}_i$*.*

Theorem 8. ([3, 4]). *Let* HE *be a bootstrappable scheme that is also weakly circular secure. Then there is a fully homomorphic encryption scheme as per Denition 3.*

References

[1] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *STOC*, pages 8493. ACM, 2005.

[2] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. J. ACM, 56(6), 2009.

[3] Craig Gentry. *A fully homomorphic encryption scheme.* PhD thesis, Stanford University, 2009. `crypto.stanford.edu/craig`.

[4] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *STOC*, pages 169178. ACM, 2009.

[5] Zvika Brakerski and Vinod Vaikuntanathan. 2011. Efficient fully homomorphic encryption from (standard) LWE. In *Proceedings of FOCS*. 97106.