CS 7880 Graduate Cryptography December 4, 2017

Lecture 24: Bootstrapping FHE, Lattice Trapdoors, Signs, IDBE

Lecturer: Daniel Wichs *Scribe: Vikrant Singhal*

1 Topics Covered

- Bootstrapping Fully Homomorphic Encryption (FHE)
- Lattice Trapdoors
- Signatures
- Identity-Based Encryption (IDBE)

2 Bootstrapping Fully Homomorphic Encryption (FHE)

The scheme we talked about in the last lecture (Leveled FHE), has a drawback, that is, we need to know the size of the circuit we would be evaluating on the cyphertexts before we set the parameters for encryption. We want to overcome this limitation to get a true FHE. This process of converting the leveled scheme to a true FHE scheme is called, "Bootstrapping."

2.1 Additional Assumption

We assume that encrypting the secret key via the corresponding public key, and making the resulting cyphertext public, is secure. We call this assumption, the "Circular Security Assumption." The new scheme would be secure under this assumption.

2.2 Structure

For this, the same encryption, decryption, and key-generation procedures are used as in leveled FHE. Suppose we have a decryption circuit for each cyphertext that we want use (as inputs and at intermediate levels) to evaluate the target circuit, with the respective cyphertext hardcoded in it. These circuits take in the secret key as the input, and output the decryptions of their respective hardcoded cyphertexts. We assume that the depth of each decryption circuit is d_{DEC}, and the maximum depth of circuits, the noise caused by which, the decryption procedure of leveled FHE can tolerate, is $d_{DEC}+1$. Let the depth of the target circuit be d.

Now, suppose the original cyphertexts have gone through the target circuit, and have resulted in some values at level $d_{DEC}+1$. This means that if they go through one more level of the ciruit, the decryption procedure would fail to correctly decrypt them. Let $c^* = \mathsf{Enc}_{pk}(sk)$. So, c^* has very low level of error because it is freshly encrypted. We use c^* to reduce the error at this point of the existing values at level, $d_{DEC}+1$. For a cyphertext,

c, at level, $d_{DEC}+1$, let $c' = \mathit{Eval}\left(\mathsf{Dec}(\cdot, c), c^*\right)$, that is, c' be the new cyphertext with noise reduced by one level. This Eval function is a homomorphic procedure, which uses the encrypted secret key to somehow decrypt c (not entirely though), and re-encrypt it to get c' that has slightly reduced noise.

Now, all values at level, $d_{DEC}+1$, have their respective noises reduced by one level. So, they can go through one more level of the circuit. Once, that happens, we employ the above procedure again. We could repeat this process until we reach the top level of the circuit.

Correctness still holds because of the maintenance of the noise levels in cyphertexts. This is almost FHE, but the question is whether it is secure to release the encryption of the secret key under the corresponding public key in public. In this case, we don't know if there is any attack against this scheme, but we also don't have a proof that this action is secure. We simply assume that the Circular Security Assumption holds.

2.3 Slightly Weaker FHE

We introduce a variant of the above scheme that is less powerful, but does not require the Circular Security Assumption. The caveats are that we need to know the size of the target circuit prior to setting the parameters, and that the size of the public key grows with the size of the target circuit. But the advantage is that the noise of the final cyphertext is the same as the noise at level, $d_{DEC}+1$.

We calculate $c_1^* = \mathsf{Enc}_{pk_2}(sk_1), c_2^* = \mathsf{Enc}_{pk_3}(sk_2), \ldots, c_d^* = \mathsf{Enc}_{pk_{d+1}(sk_d)}$. Then $sk^* = sk_d$ and $pk^* = (pk_1, c_1^*, \ldots, c_d^*)$. So, we perform $\mathit{Eval}\left(\mathsf{Dec}(\cdot, c), c_1^*\right)$ to get encryption of the values under pk_2, and so on. We get encryptions under a new public key at each level.

3 Lattice Trapdoors

We showed earlier that the SIS problem could be solved when we had a special gadget matrix, G, but was hard when the matrix was chosen randomly instead. Now, we want to come up with a matrix, A, along with a trapddoor, td, such that we could solve SIS efficiently, given that trapdoor.

$$\overline{A} \leftarrow \mathbb{Z}_q^{n \times \frac{m}{2}} \qquad\qquad R \leftarrow \{0,1\}^{\frac{m}{2} \times \frac{m}{2}}$$

$$A = \left[\overline{A} \mid \overline{A} \cdot R + G\right]$$

$$td = R$$

For the SIS problem, we want to solve $Au = v$, where u is $m \times 1$ dimensional. We write u as a concatenation of two $\frac{m}{2} \times 1$ vectors, u_1 and u_2.

We claim that the distribution of A is statistically close to $\mathsf{U}^{n \times m}$. Since $h_{\overline{A}}(R) = \overline{A} \cdot R$ is random for a random $\overline{A}$ (using Universal Hash Lemma), A is also random.

Since we know that SIS is hard when A is random, we have hardness when R is unknown.

Now, we try to solve $Au = v$ to obtain a "short" u.

$$\begin{aligned} Au &= v \\ \Rightarrow \overline{A}u_1 + (\overline{A} \cdot R + G)u_2 &= v \\ \Rightarrow \overline{A}(u_1 + Ru_2) + Gu_2 &= v \end{aligned}$$

Let $u^* = u_1 + Ru_2$. Sample u^* randomly from $\{0,1\}^{\frac{m}{2}}$. Note that this u^* is "short". We have the following.

$$\begin{aligned} u_2 &= G^{-1}(v - \overline{A}u^*) \in \{0,1\}^{\frac{m}{2}} \\ \Rightarrow u_1 &= u^* - Ru_2 \end{aligned}$$

We make two observations here. Firstly, to obtain u_1 from u^*, we had to make use of the trapdoor, R. Secondly, $\|u_1\|_\infty \leq \frac{m}{2} + 1$.

However, we cannot really use this procedure because this method of obtaining u_1 leaks information about R. So, if the adversary makes sufficient number of queries, he may be able to learn R. Therefore, we need an additional property to maintain security of the trapdoor above. Suppose $(A, td) \leftarrow \mathsf{Gen}(1^n)$, $v \leftarrow \mathbb{Z}_q^n$ and $u \leftarrow \mathsf{Inv}_{td}(v)$. Then we want the following to hold.

$$(A, u) \approx (A', u')$$

Where $A' \leftarrow \mathbb{Z}_q^{n\times m}$ and $u' \leftarrow \chi^m$ (χ being an efficiently samplable distribution). This overcomes the limitation posed by the aforementioned trapdoor scheme. There exists a trapdoor scheme that satisfies this property too, but we don't discuss it over here.

4 Signatures

We use the trapdoor scheme that we didn't discuss (though it works) to construct a signature scheme.

$$\begin{aligned} \mathsf{KeyGen}(1^n) :& (A, td) \leftarrow \mathsf{Gen}(1^n) \\ & pk = A, sk = td \\ \mathsf{Sign}_s k(m) :& v = RO(m) \in \mathbb{Z}_q^n \\ & u \leftarrow \mathsf{Inv}_{td}(v) \\ \mathsf{Verify}_{pk}(m, u) :& \text{Check if } Au = RO(m) \\ & \text{and if } u \text{ is "short".} \end{aligned}$$

Note that u has to be short enough for SIS to be hard, but long enough that it looks like a sample from the samplable distribution.

To prove its security, a reduction from SIS is needed. The proof goes the same way as the proof for signatures using RSA (under RO), and utilizes the property mentioned in the previous section.

5 Identity-Based Encryption (IDBE)

Consider the situation, where many parties, who want to privately communicate with each other, exist, and they get their private and public keys from a common head-organization. It may be infeasible for the organization to distribute all the keys right at the beginning. In this setting, IDBE comes into play, as it helps create keys for parties, based on their unique respective identites, and encrypt/decrypt messages according to those keys. The main idea behind the structure of such schemes is that the organization generates a master public key

(mpk) and a master secret key (msk), which are then used to generate private keys for all parties based on their identities.

5.1 Structure

More formally, we have the following structure.

$$\begin{aligned} (mpk, msk) &\leftarrow \mathsf{Setup}(1^n) \\ sk_{id} &\leftarrow \mathsf{KeyGen}(1^n, id, msk) \\ ct &\leftarrow \mathsf{Enc}_{mpk}(id, m) \\ m &\leftarrow \mathsf{Dec}_{sk_{id}}(ct) \end{aligned}$$

5.2 Correctness

We require the following notion of correctness to hold.

$$\mathbb{P}\left[\mathsf{Dec}_{sk_{id}}\left(\mathsf{Enc}_{mpk}(id, m)\right) = m \mid (mpk, msk) \leftarrow \mathsf{Setup}(1^n), sk_{id} \leftarrow \mathsf{KeyGen}(1^n, id, msk)\right] = 1$$

5.3 Security

We define the following security game ($\mathsf{IND\text{-}ID\text{-}CPAGame}^b$).

$$\begin{array}{lcl}
\mathcal{A} & & \mathcal{C}(b) \\
 & & (mpk, msk) \leftarrow \mathsf{KeyGen}(1^n) \\
 & \xleftarrow{\qquad\qquad mpk \qquad\qquad} & \\
\text{(I)} & \xrightarrow{\qquad\qquad id_i \qquad\qquad} & \\
 & \xleftarrow{\qquad\qquad sk_{id_i} \qquad\qquad} & \\
 & \xrightarrow{id^* \text{ st. } id^* \notin \{id_i\}, m_0, m_1} & \\
 & \xleftarrow{ct \leftarrow \mathsf{Enc}_{mpk}(id^*, m_b)} & \\
\text{(II)} & \xrightarrow{\qquad id_j \text{ st. } id_j \neq id^* \qquad} & \\
 & \xleftarrow{\qquad\qquad sk_{id_j} \qquad\qquad} & \\
\downarrow b' \in \{0,1\} & &
\end{array}$$

We say that the scheme is $\mathsf{IND\text{-}ID\text{-}CPA}$ secure if,

$$\mathsf{IND\text{-}ID\text{-}CPAGame}^0(n) \approx \mathsf{IND\text{-}ID\text{-}CPAGame}^1(n).$$

5.4 Scheme

We assume again that we have that good trapdoor scheme that we didn't talk about in the lecture. We have the following.

$$\begin{aligned}
\mathsf{KeyGen}(1^n) :& (A, td) \leftarrow \mathsf{Gen}(1^n) \\
& mpk = A, msk = td \\
& \vec{v} = RO(id) \\
& sk_{id} = \vec{u} \leftarrow \mathsf{Inv}_{td}(\vec{v}) \\
\mathsf{Enc} :& \vec{v} = RO(id) \\
& \vec{b} \leftarrow \vec{s}A + \vec{e} \\
& w \leftarrow \langle \vec{s}, \vec{v} \rangle + e' + m \left\lfloor \frac{q}{2} \right\rfloor \\
& ct = (\vec{b}, w) \\
\mathsf{Dec} :& m = \text{Round}\left(w - \left\langle \vec{b}, \vec{u} \right\rangle \right)
\end{aligned}$$

Correctness follows from that of Regev's proof. We get,

$$\text{Round}\left(w - \left\langle \vec{b}, \vec{u} \right\rangle \right) = \text{Round}\left(e' - \langle \vec{e}, \vec{u} \rangle + m \left\lfloor \frac{q}{2} \right\rfloor \right).$$

So, if we set the parameters correctly, we get the required result.