CSE 40622 Cryptography
Writing Assignment 09 (Lecture 18-20)

Name: Walker Bagley (wbagley)

1. (10 pts) In the somewhat homomorphic symmetric-key encryption, suppose we have the following parameters:

   - sk's bitwidth is $bit(\mathsf{sk})$ bits.
   - r's bitwidth is $bit(r)$ bits.
   - q's bitwidth is $bit(q)$ bits.

   What is the multiplicative depth of the scheme with these parameters?

   Answer:

   Recall that the ciphertext $c = pq + 2r + m$ where $bit(p) = bit(\mathsf{sk})$. We know that the bitwidth of the noise $bit(2r)$ must be less than $bit(p)$ in order for decryption to work. We also know that $bit(2r)$ doubles in the worst case each time we multiply. Also, $bit(2r) = 1 + bit(r)$. So $\log_2(bit(\mathsf{sk}))$ would give the multiplicative depth if our noise has a bit length of 1. However, out noise has a bit length of $bit(r)+1$, and thus there are $\log_2(bit(r)+1)$ multiplications we cannot do due to our noise already being a certain size. So, subtracting gives us a multiplicative depth of $\log_2(bit(\mathsf{sk})) - \log_2(bit(r)+1) = \log_2(\frac{bit(\mathsf{sk})}{bit(r)+1})$. This may not be an integer, so we floor this result as we cannot have a partial multiplication. Thus, the multiplicative depth of this scheme would be $\lfloor \log_2(\frac{bit(\mathsf{sk})}{bit(r)+1}) \rfloor$.

2. (20 pts) Suppose there are two valid ciphertexts $ct_1 = (\mathbf{v}_1(-\mathbf{as} + \mathbf{e}) + \mathbf{e}_{1,0} + \mathbf{m}_1, \mathbf{v}_1\mathbf{a} + \mathbf{e}_{1,1})$ and $ct_2 = (\mathbf{v}_2(-\mathbf{as} + \mathbf{e}) + \mathbf{e}_{2,0} + \mathbf{m}_2, \mathbf{v}_2\mathbf{a} + \mathbf{e}_{2,1})$, which are the ciphertexts of $\mathbf{m}_1$ and $\mathbf{m}_2$ respectively. We have learned that the outcome of homomorphic addition on these two ciphertexts is $c_{add} = (ct_1[0] + ct_2[0], ct_1[1] + ct_2[1])$. Apply the decryption algorithm on c_{add}, simplify the terms (i.e., show all the equations), and show that the outcome of the decryption will be approximately $\mathbf{m}_1 + \mathbf{m}_2$.

   Answer:

   The decryption algorithm on c_{add} would be equal to

$$\begin{aligned}
c_{add}[0] + c_{add}[1]\mathbf{s} &= ct_1[0] + ct_2[0] + (ct_1[1] + ct_2[1])\mathbf{s} \\
&= \mathbf{v}_1(-\mathbf{as} + \mathbf{e}) + \mathbf{e}_{1,0} + \mathbf{m}_1 + \mathbf{v}_2(-\mathbf{as} + \mathbf{e}) + \mathbf{e}_{2,0} + \mathbf{m}_2 + (\mathbf{v}_1\mathbf{a} + \mathbf{e}_{1,1} + \mathbf{v}_2\mathbf{a} + \mathbf{e}_{2,1})s \\
&= (\mathbf{v}_1 + \mathbf{v}_2)(-\mathbf{as} + \mathbf{e}) + (\mathbf{e}_{1,0} + \mathbf{e}_{2,0}) + \mathbf{m}_1 + \mathbf{m}_2 + (\mathbf{v}_1\mathbf{a} + \mathbf{e}_{1,1} + \mathbf{v}_2\mathbf{a} + \mathbf{e}_{2,1})s \\
&= (\mathbf{v}_1 + \mathbf{v}_2)\mathbf{e} + (\mathbf{e}_{1,0} + \mathbf{e}_{2,0}) + \mathbf{m}_1 + \mathbf{m}_2 + (\mathbf{e}_{1,1} + \mathbf{e}_{2,1})s \\
&\approx (\mathbf{e}_{1,0} + \mathbf{e}_{2,0}) + \mathbf{m}_1 + \mathbf{m}_2 + (\mathbf{e}_{1,1} + \mathbf{e}_{2,1})s \\
&\approx \mathbf{m}_1 + \mathbf{m}_2 + (\mathbf{e}_{1,1} + \mathbf{e}_{2,1})s \\
&\approx \mathbf{m}_1 + \mathbf{m}_2
\end{aligned}$$

   Since $v_1, v_2, e, e_{1,0}, e_{2,0}, e_{1,1}, e_{2,1}, s$ are all very small in comparison to m_1, m_2.

3. (20 pts) At the end of a homomorphic multiplication of a cipher of $\mathbf{m}_1$ and a cipher of $\mathbf{m}_2$, why do we wish to have a cipher of $\left\lfloor \frac{\mathbf{m}_1\mathbf{m}_2}{\Delta} \right\rceil$ instead of $\mathbf{m}_1\mathbf{m}_2$?

   Answer:

   Remember that we multiply m by Δ during encoding before encrypting so that the coefficients of m dominate the scheme. We also divide the decryption by Δ when we decode. Let m_1, m_2 designate the original messages, so we encrypt and decrypt using $\Delta m_1, \Delta m_2$. Then $\mathbf{m}_1\mathbf{m}_2 = \Delta^2 m_1 m_2$. We just want $m_1 m_2$ by the end and will divide the decryption by Δ during decoding, so we must get rid of the other Δ. We do this by dividing by Δ once again in the cipher, so that upon decryption, we are left with $\Delta m_1 m_2$ which will reduce to $m_1 m_2$ on decoding, which is exactly what we want.

4. Let's turn the somewhat homomorphic symmetric-key encryption in Section 2.1.1 to a fully homomorphic symmetric-key encryption.

   * Recall that $Enc_{\mathsf{sk}}(x)$ is a ciphertext that can be used for homomorphic operations even though x is not binary. In other words, $Enc_{\mathsf{sk}}(x)+Enc_{\mathsf{sk}}(y) = Enc_{\mathsf{sk}}(x+y)$ and $Enc_{\mathsf{sk}}(x)\cdot Enc_{\mathsf{sk}}(y) = Enc_{\mathsf{sk}}(x\cdot y)$ even though x, y are not binary.

   4.1. (10 pts) Describe the decryption function with arithemtic operations (i.e., $\pm, \times, \div$) and the floor function only.

   Answer:

   Recall that the decryption scheme is $m = c \mod p \mod 2$, and we can write $a \mod b$ as $a - b \times \lfloor a \div b \rfloor$. So we can rewrite the decryption scheme as follows:

$$
\begin{aligned}
c \mod p &= a = c - p \times \lfloor c \div p \rfloor \\
m = a \mod 2 &= a - 2 \times \lfloor a \div 2 \rfloor \\
m &= c - p \times \lfloor c \div p \rfloor - 2 \times \lfloor c - p \times \lfloor c \div p \rfloor \div 2 \rfloor
\end{aligned}
$$

   4.2. (30 pts) Let $Enc_{\mathsf{sk}}(x)$ be the encryption of x with the secret key sk. Suppose we have a fresh ciphertext of sk (i.e., $Enc_{\mathsf{sk}}(\mathsf{sk})$), and we also have an encryption oracle $\mathcal{O}_{enc}$ which returns a fresh ciphertext of x given the input x. In other words, $\mathcal{O}_{enc}(x) = Enc_{\mathsf{sk}}(x)$. Finally, suppose we have another oracle $\mathcal{O}_{floor}$ which returns a fresh ciphertext of $\lfloor x/y \rfloor$ given a ciphertext of x and a ciphertext of y. In other words, $\mathcal{O}_{floor}(Enc_{\mathsf{sk}}(x), Enc_{\mathsf{sk}}(y)) = Enc_{\mathsf{sk}}(\lfloor x/y \rfloor)$.

   - The ciphertexts of x, y may contain large noises, but the noise should not be too large and the ciphertexts of x, y should be decryptable.

   Describe the homomorphic operations needed for bootstrapping a ciphertext c with the encrypted secret key and the two oracles above.

   Answer:

   Use $\mathcal{O}_{enc}$ to calculate $\mathcal{O}_{enc}(c) = Enc_{\mathsf{sk}}(c), \mathcal{O}_{enc}(p) = Enc_{\mathsf{sk}}(p), \mathcal{O}_{enc}(2) = Enc_{\mathsf{sk}}(2)$ and remember that $c = Enc_{\mathsf{sk}}(m)$. Then use $\mathcal{O}_{floor}$ to calculate $\mathcal{O}_{floor}(Enc_{\mathsf{sk}}(c), Enc_{\mathsf{sk}}(p)) = Enc_{\mathsf{sk}}(\lfloor c/p \rfloor)$.

   Lets start with $\alpha = c - p \times \lfloor c/p \rfloor$, which will look like this after applying the homomorphisms given by Enc_{sk}:

$$
\begin{aligned}
\alpha &= Enc_{\mathsf{sk}}(c) - Enc_{\mathsf{sk}}(p) \times Enc_{\mathsf{sk}}(\lfloor c/p \rfloor) \\
&= Enc_{\mathsf{sk}}(c) - Enc_{\mathsf{sk}}(p \times \lfloor c/p \rfloor) \\
&= Enc_{\mathsf{sk}}(c - p \times \lfloor c/p \rfloor)
\end{aligned}
$$

   Then, to finish the decryption, we convert $m = \alpha - 2 \times \lfloor \alpha/2 \rfloor$:

$$
\begin{aligned}
m &= \alpha - Enc_{\mathsf{sk}}(2) \times \lfloor \alpha / Enc_{\mathsf{sk}}(2) \rfloor \\
&= Enc_{\mathsf{sk}}(c - p \times \lfloor c/p \rfloor) - Enc_{\mathsf{sk}}(2) \times \lfloor Enc_{\mathsf{sk}}(c - p \times \lfloor c/p \rfloor) / Enc_{\mathsf{sk}}(2) \rfloor
\end{aligned}
$$

   Use $\mathcal{O}_{floor}$ to calculate $\mathcal{O}_{floor}(Enc_{\mathsf{sk}}(c - p \times \lfloor c/p \rfloor), Enc_{\mathsf{sk}}(2)) = Enc_{\mathsf{sk}}(\lfloor c - p \times \lfloor c/p \rfloor / 2 \rfloor)$, so our encryption will now look like this after once again applying the homomorphism properties of Enc_{sk}:

$$
\begin{aligned}
m &= Enc_{\mathsf{sk}}(c - p \times \lfloor c/p \rfloor) - Enc_{\mathsf{sk}}(2) \times Enc_{\mathsf{sk}}(\lfloor c - p \times \lfloor c/p \rfloor / 2 \rfloor) \\
&= Enc_{\mathsf{sk}}(c - p \times \lfloor c/p \rfloor) - Enc_{\mathsf{sk}}(2 \times \lfloor c - p \times \lfloor c/p \rfloor / 2 \rfloor) \\
m &= Enc_{\mathsf{sk}}(c - p \times \lfloor c/p \rfloor - 2 \times \lfloor c - p \times \lfloor c/p \rfloor / 2 \rfloor)
\end{aligned}
$$

Which we can then decrypt to find m.

4.3. (10 pts) How much multiplicative depth do we need for the bootstrapping above?

Answer:

Since we use the oracles to generate fresh encryptions for everything, any operations can be performed without significantly increasing the noise. As a result, we only need a multiplicative depth of 1 to perform the bootstrapping method above.