CS-E4340: Cryptography **I-II 2022/2023**

Exercise 8: Fully Homomorphic Encryption

Deadline: 11:30 on November 14, 2022 via MyCourses as a single pdf file

Abstract

This exercise is designed to help students to ...

- understand noise growth in fully homomorphic encryption constructions, and
- get a feeling of how fully homomorphic encryption can be applied in different scenarios.

For this exercise sheet, it suffices to provide asymptotic bounds, e.g. in Big-O notation.

Question 1 (Noise Growth)**.** In this question, we study the noise growth of ciphertexts in the FHE construction of Gentry, Sahai, and Waters (GSW), and the effect of choosing different circuit representations of a function during homomorphic evaluation. First, we recall the GSW construction for $n, q \in \mathbb{N}$, $m = n \cdot (\lfloor \log q \rfloor + 1)$, and χ being the uniform distribution over $\mathbb{Z}_\beta$ for some $\beta \in \mathbb{N}$.

$\underline{\mathsf{KGen}(1^\lambda)}$

$\overline{\mathbf{A}} \leftarrow\!\$\ \mathbb{Z}_q^{(n-1)\times m}$

$\mathbf{s} \leftarrow\!\$\ \mathbb{Z}_q^{n-1}$

$\mathbf{e} \leftarrow\!\$\ \chi^m$

$\mathbf{b}^\mathsf{T} \coloneqq \mathbf{s}^\mathsf{T} \cdot \overline{\mathbf{A}} + \mathbf{e}^\mathsf{T} \bmod q$

$\mathbf{A} \coloneqq \begin{pmatrix} \overline{\mathbf{A}} \\ \mathbf{b}^\mathsf{T} \end{pmatrix} \in \mathbb{Z}_q^{n\times m}$

$(\mathsf{pk}, \mathsf{sk}) \coloneqq (\mathbf{A}, \mathbf{s})$

return $(\mathsf{pk}, \mathsf{sk})$

$\underline{\mathsf{Enc}(\mathsf{pk}, x \in \{0,1\})}$

$\mathbf{R} \leftarrow\!\$\ \chi^{m\times m}$

$\mathsf{ctxt} \coloneqq \mathbf{C} \coloneqq \mathbf{A} \cdot \mathbf{R} + x \cdot \mathbf{G} \bmod q$

return ctxt

$\underline{\mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt})}$

$\overline{\mathbf{x}}^\mathsf{T} \coloneqq (-\mathbf{s}^\mathsf{T}, 1) \cdot \mathbf{C} \bmod q$

$\bar{x} \coloneqq$ last entry of $\overline{\mathbf{x}}$

if $|\bar{x}| < q/4$ **then**

  return 0

else

  return 1

$\underline{\mathsf{Eval}(\mathsf{pk}, +, \mathsf{ctxt}_0, \mathsf{ctxt}_1)}$

$\mathsf{ctxt} \coloneqq \mathbf{C} \coloneqq \mathbf{C}_0 + \mathbf{C}_1 \bmod q$

return ctxt

$\underline{\mathsf{Eval}(\mathsf{pk}, \times, \mathsf{ctxt}_0, \mathsf{ctxt}_1)}$

$\mathsf{ctxt} \coloneqq \mathbf{C} \coloneqq \mathbf{C}_0 \cdot \mathbf{G}^{-1}(\mathbf{C}_1) \bmod q$

return ctxt

As explained in the lecture notes, homomorphic evaluation maintains the invariant that a ciphertext $\mathbf{C}_x$ encrypting a value x is of the form $\mathbf{C}_x = \mathbf{A} \cdot \mathbf{R}_x + x \cdot \mathbf{G} \bmod q$. In the following, consider the setting where $\ell = 2^k$ is a power of 2, $x_i \in \{0,1\}$, and $\mathbf{C}_{x_i} \in \mathsf{Enc}(\mathsf{pk}, x_i)$, for each $i \in [\ell]$. The noise level of $\mathbf{C}_x$ is defined as the maximum of the absolute values of the entries of $\mathbf{R}_x$ which, with a slight abuse of notation, could be denoted by $\|\mathbf{R}_x\|$. Given a circuit Γ, we denote by $\mathbf{C}_{\Gamma(x_1,\ldots,x_\ell)}$ the ciphertext obtained by homomorphically evaluating Γ over $(\mathbf{C}_{x_1}, \ldots, \mathbf{C}_{x_\ell})$.

(a) Write down the norm of the noise term $\mathbf{R}_{x_1 \cdot x_2}$, i.e. $\|\mathbf{R}_{x_1 \cdot x_2}\|$, obtained when homomorphically computing $x_1 \cdot x_2$, in terms of $\|\mathbf{R}_{x_1}\|$ and $\|\mathbf{R}_{x_2}\|$. How does $\|\mathbf{R}_{x_1 \cdot x_2}\|$ scale with $\|\mathbf{R}_{x_1}\|$ and $\|\mathbf{R}_{x_2}\|$? For example, does it depend more on $\mathbf{R}_{x_1}$ (or $\mathbf{R}_{x_2}$) and by how much, or equally on $\mathbf{R}_{x_1}$ and $\mathbf{R}_{x_2}$?

(b) Consider the function $f(x_1, \ldots, x_\ell) = \prod_{i=1}^{\ell} x_i$ where the multiplication is done over $\mathbb{Z}$. To compute f, conventional wisdom suggests to perform multiplications in a tree-like fashion, i.e. first compute

$x_1 \cdot x_2, x_3 \cdot x_4, \ldots, x_{\ell-1} \cdot x_\ell$, then compute $\prod_{i=1}^{4} x_i, \prod_{i=5}^{8} x_i, \ldots, \prod_{i=\ell-3}^{\ell} x_i$, and so on, until obtaining $\prod_{i=1}^{\ell} x_i$. Let $\Gamma_{\texttt{tree}}$ denote such a circuit.

Write down a tight upper bound of the noise level of $\mathbf{C}_{\Gamma_{\texttt{tree}}(x_1,\ldots,x_\ell)}$ in terms of (ℓ, n, q, β).

(c) (Optional) Design a circuit Γ for $f(x_1, \ldots, x_\ell) = \prod_{i=1}^{\ell} x_i$ such that the noise level of $\mathbf{C}_{\Gamma(x_1,\ldots,x_\ell)}$ is linear in ℓ. Write down a tight upper bound of the noise level of $\mathbf{C}_{\Gamma(x_1,\ldots,x_\ell)}$ in terms of (ℓ, n, q, β).

(a) Let $\mathbf{C}_{x_1}$ and $\mathbf{C}_{x_2}$ be the ciphertexts of x_1 and x_2 respectively, and let $\mathbf{C}_{x_1 \cdot x_2}$ be the resulting ciphertext when homomorphically evaluating $x_1 \cdot x_2$. We have

$$\begin{aligned}
\mathbf{C}_{x_1 \cdot x_2} =& \mathbf{C}_{x_1} \cdot \mathbf{G}^{-1}(\mathbf{C}_{x_2}) \bmod q \\
=& (\mathbf{A}\mathbf{R}_{x_1} + x_1\mathbf{G}) \cdot \mathbf{G}^{-1}(\mathbf{C}_{x_2}) \bmod q \\
=& \mathbf{A}\mathbf{R}_{x_1}\mathbf{G}^{-1}(\mathbf{C}_{x_2}) + x_1(\mathbf{A}\mathbf{R}_{x_2} + x_2\mathbf{G}) \bmod q \\
=& \mathbf{A}\left(\mathbf{R}_{x_1}\mathbf{G}^{-1}(\mathbf{C}_{x_2}) + x_1\mathbf{R}_{x_2}\right) + x_1x_2\mathbf{G} \bmod q.
\end{aligned}$$

The noise term of $\mathbf{C}_{x_1 \cdot x_2}$ is therefore

$$\mathbf{R}_{x_1 \cdot x_2} = \mathbf{R}_{x_1}\mathbf{G}^{-1}(\mathbf{C}_{x_2}) + x_1\mathbf{R}_{x_2}.$$

Since $\mathbf{G}^{-1}(\mathbf{C}_{x_2}) \in \{0,1\}^{m \times m}$, we have

$$\|\mathbf{R}_{x_1 \cdot x_2}\| \leq m \cdot \|\mathbf{R}_{x_1}\| + \|\mathbf{R}_{x_2}\| = m \cdot \frac{\beta}{2} + \frac{\beta}{2} = (m+1) \cdot \frac{\beta}{2} = O(\beta \cdot n \cdot \log q).$$

Notice that in the above, $\|\mathbf{R}_{x_1}\|$ is multiplied by $m \approx n \cdot \log q$, therefore $\|\mathbf{R}_{x_1 \cdot x_2}\|$ depends roughly $n \cdot \log q$ times more on $\|\mathbf{R}_{x_1}\|$ than on $\|\mathbf{R}_{x_2}\|$.

(b) From the result in Part (a), we know that evaluating each gate in $\Gamma_{\texttt{tree}}$ leads to a noise growth of a factor of $(m+1)$. Since $\Gamma_{\texttt{tree}}$ has depth $\log \ell$, the noise level of the output ciphertext is

$$(m+1)^{\log \ell} \cdot \frac{\beta}{2} = \ell^{O(\log n + \log\log q)} \cdot \beta,$$

which is super-polynomial in ℓ.

(c) Let Γ be as follows: in the first layer we compute $y_1 = x_1 \cdot x_2$, and in each subsequent layer $i = 2, \ldots, \ell - 1$, we compute $y_i = y_{i-1} \cdot x_{i+1}$. That is, in each layer of the circuit we multiply one input x_i to the output of the previous layer.

Using again the result from Part (a), we have that in each layer of evaluation, we only add $m\frac{\beta}{2}$ to the noise level of the ciphertext. Therefore, the final ciphertext has a noise level of

$$\frac{\beta}{2} + (\ell - 1)m\frac{\beta}{2} = O(\beta \cdot \ell \cdot n \cdot \log q),$$

which is linear in ℓ.

Question 2 (Private Information Retrieval from FHE)**.** In this question, we study a classic application of FHE to construct private information retrieval (PIR) schemes.

First, we introduce the notion of PIR. In the setting of (single-server) PIR there is one client and one server.

The server holds a large database $D = (d_i)_{i=1}^N \in \{0,1\}^N$ of size N (imagine N to be around 2^{30} to 2^{50}) and the client is interested to learn the value of d_i for some secret position $i \in [N]$.

Definition (Private Information Retrieval (PIR))**.** A PIR scheme consists of a tuple of PPT algorithms $\mathsf{PIR}.(\mathsf{Query}, \mathsf{Eval}, \mathsf{Recover})$ with the following syntax:

- $(\mathsf{qry}, \mathsf{rec}) \leftarrow \mathsf{PIR.Query}(1^\lambda, i \in [N])$: On input a position $i \in [N]$ generate a query qry and some recovery information rec.
- $\mathsf{rsp} \leftarrow \mathsf{PIR.Eval}(D, \mathsf{qry})$: On input a database $D \in \{0,1\}^N$ and a query qry return a response rsp.
- $d \leftarrow \mathsf{PIR.Recover}(\mathsf{rec}, \mathsf{rsp})$: From some recovery information rec and a response rsp, recover a database entry $d \in \{0,1\}$.

A PIR scheme is correct if for any $\lambda, N \in \mathbb{N}$, any database $D \in \{0,1\}^N$, any position $i \in [N]$, any query and recovery information $(\mathsf{qry}, \mathsf{rec}) \in \mathsf{PIR.Query}(1^\lambda, i)$, and any response $\mathsf{rsp} \in \mathsf{PIR.Eval}(D, \mathsf{qry})$, it holds that $\mathsf{PIR.Recover}(\mathsf{rec}, \mathsf{rsp}) = d_i$.

A PIR scheme PIR is secure if for any (two-stage) PPT adversary $\mathcal{A}$ and any $N = \mathsf{poly}(\lambda)$ it holds that

$$\left|\Pr\left[\mathsf{PIRSecurity}^0_{\mathsf{PIR},\mathcal{A}}(1^\lambda, 1^N) = 1\right] - \Pr\left[\mathsf{PIRSecurity}^1_{\mathsf{PIR},\mathcal{A}}(1^\lambda, 1^N) = 1\right]\right| \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{PIRSecurity}^b_{\mathsf{PIR},\mathcal{A}}$ for $b \in \{0,1\}$ is defined as follows:

$\mathsf{PIRSecurity}^b_{\mathsf{PIR},\mathcal{A}}(1^\lambda, 1^N)$

$(i_0, i_1) \leftarrow \mathcal{A}(1^\lambda, 1^N)$

$(\mathsf{qry}^*, \mathsf{rec}^*) \leftarrow \mathsf{PIR.Query}(1^\lambda, i_b)$

$b' \leftarrow \mathcal{A}(\mathsf{qry}^*)$

return b'

To construct a PIR scheme PIR from an FHE scheme FHE with message space $\mathcal{X} \subseteq \mathbb{Z}$, the idea is as follows.

Construction. First, we design an encoding function

$$\begin{aligned} \mathsf{Encode} : [N] &\to \mathcal{X}^\ell \\ i &\mapsto (x_{i,1}, \ldots, x_{i,\ell}) \end{aligned}$$

which encodes $i \in [N]$ as a tuple of $(x_{i,1}, \ldots, x_{i,\ell}) \in \mathcal{X}^\ell$. Next, for any given database $D \in \{0,1\}^N$, we design an ℓ-variate polynomial $f_D(X_1, \ldots, X_\ell)$ with coefficients in $\mathbb{Z}$ satisfying

$$f_D(\mathsf{Encode}(i)) = f_D(x_{i,1}, \ldots, x_{i,\ell}) = d_i$$

for all $i \in [N]$. To query position i, the client samples a fresh FHE key pair $(\mathsf{pk}, \mathsf{sk})$, encrypts $(x_{i,1}, \ldots, x_{i,\ell})$ under the public key pk to give $(\mathsf{ctxt}_1, \ldots, \mathsf{ctxt}_\ell)$, i.e.

$$\mathsf{ctxt}_j \leftarrow \mathsf{FHE.Enc}(\mathsf{pk}, x_{i,j}),$$

and sends $(\mathsf{pk}, \mathsf{ctxt}_1, \ldots, \mathsf{ctxt}_\ell)$ to the server. The server then homomorphically evaluates f_D on the ciphertexts, i.e.

$$\mathsf{ctxt} \leftarrow \mathsf{FHE.Eval}(\mathsf{pk}, f_D, \mathsf{ctxt}_1, \ldots, \mathsf{ctxt}_\ell),$$

and returns the resulting ciphertext ctxt to the client. The client then recovers $d_i \leftarrow \mathsf{FHE.Dec}(\mathsf{sk}, \mathsf{ctxt})$.

(a) Write down an unconditionally secure (i.e. without using FHE or any other cryptographic primitives or assumptions) PIR scheme with communication complexity $O(N)$, i.e. $|\mathsf{qry}| + |\mathsf{rsp}| = O(N)$. [Hint: The construction is really simple.]

(b) Assuming the functionalities of Encode and f_D as outlined above, show that the above construction of PIR from FHE is

  (i) correct if the FHE is correct, and

  (ii) secure if the FHE is IND-CPA-secure. [Hint: Define ℓ hybrid security experiments to swap out the ciphertexts $\mathsf{ctxt}_1, \ldots, \mathsf{ctxt}_\ell$ one by one.]

(c) Suppose that an FHE public key pk is of size at most $p(\lambda)$, and an FHE ciphertext ctxt_x of any message $x \in \mathcal{X}$ is of size at most $p(\lambda) \cdot \log |\mathcal{X}|$, for some fixed polynomial $p(\lambda) \in \mathsf{poly}(\lambda)$. What is the asymptotic communication complexity, i.e. $|\mathsf{qry}| + |\mathsf{rsp}|$, in terms of (N, λ), of the PIR construction outlined above for the following choices of ℓ, $\mathcal{X}$, Encode and f_D?

  (i) Without loss of generality, assume that $N = n^2$ for some $n \in \mathbb{N}$. Pack the entries of D into a square matrix $\mathbf{M} = (m_{i,j})_{i,j} \in \{0,1\}^{n \times n}$ so that $m_{h,k} = d_{(h-1)\cdot n + k}$. Let $\ell = 2$ and $\mathcal{X} = \{0,1\}^n$. Define $\mathsf{Encode}(i) = (\mathbf{e}_h, \mathbf{e}_k) \in \mathcal{X}^2$, i.e. the h-th and k-th unit vectors, where $i = (h-1) \cdot n + k$ with $h, k \in [n]$. Define the degree-2 polynomial $f_D(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\intercal \cdot \mathbf{M} \cdot \mathbf{y}$.

  (ii) Without loss of generality, assume that $N = \binom{\ell}{k}$ for some $\ell = \ell(N)$ and constant $k \in \mathbb{N}$. Let $\mathcal{X} = \{0,1\}$. Let $\mathbf{s}_i = (s_{i,1}, \ldots, s_{i,\ell}) \in \mathcal{X}^\ell$ be the (lexicographically) i-th ℓ-dimensional binary vector with exactly k-many 1's. Define $\mathsf{Encode}(i) = \mathbf{s}_i$. Define the degree-$k$ polynomial $f_D(X_1, \ldots, X_\ell) = \sum_{i \in [N]} d_i \cdot \prod_{j : s_{i,j} = 1} X_j$. [Hint: Use Stirling approximation.]

(d) (Optional) Choose ℓ and $\mathcal{X}$ and design an encoding function Encode and a polynomial f_D with the functionality outlined in the construction above, so that the above construction template yields a PIR scheme with communication complexity $\mathsf{poly}(\lambda, \log N)$.

(a) $\mathsf{PIR.Query}$ outputs $(\mathsf{qry}, \mathsf{rec}) = (\epsilon, \epsilon)$, where ϵ denotes the empty string. $\mathsf{PIR.Eval}$, on input $D \in \{0,1\}^N$ and $\mathsf{qry} = \epsilon$, outputs $\mathsf{rsp} = D$. $\mathsf{PIR.Recover}$, on input $\mathsf{rec} = \epsilon$ and $\mathsf{rsp} = D$, outputs $d = D[i]$. Clearly, $|\mathsf{qry}| + |\mathsf{rsp}| = N$.

(b) (i) Fix any $\lambda, N \in \mathbb{N}$, any database $D \in \{0,1\}^N$, any position $i \in [N]$, any query and recovery information $(\mathsf{qry}, \mathsf{rec}) \in \mathsf{PIR.Query}(1^\lambda, i)$, and any response $\mathsf{rsp} \in \mathsf{PIR.Eval}(D, \mathsf{qry})$. By construction, we have $\mathsf{qry} = (\mathsf{pk}, \mathsf{ctxt}_1, \ldots, \mathsf{ctxt}_\ell)$, $\mathsf{rec} = \mathsf{sk}$, and $\mathsf{rsp} = \mathsf{FHE.Eval}(\mathsf{pk}, f_D, \mathsf{ctxt}_1, \ldots, \mathsf{ctxt}_\ell)$, where $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{FHE.KGen}(1^\lambda)$, $\mathsf{ctxt}_j \in \mathsf{FHE.Enc}(\mathsf{pk}, x_{i,j})$ for all $j \in [\ell]$, and $(x_{i,1}, \ldots, x_{i,\ell}) = \mathsf{Encode}(i)$. By the correctness of FHE, we have

$$\mathsf{PIR.Recover}(\mathsf{rec}, \mathsf{rsp}) = \mathsf{FHE.Dec}(\mathsf{sk}, \mathsf{rsp}) = f_D(\mathsf{Encode}(i)) = d_i,$$

as desired.

(ii) We prove by a hybrid argument. Let $\mathsf{qry}^* = (\mathsf{pk}, \mathsf{ctxt}_1, \ldots, \mathsf{ctxt}_\ell)$ denote the challenge query computed in $\mathsf{PIRSecurity}^b_{\mathsf{PIR},\mathcal{A}}(1^\lambda, 1^N)$. Consider the hybrid experiments Hyb_j for $j \in \{0, \ldots, \ell\}$ where

- $\mathsf{ctxt}_1, \ldots, \mathsf{ctxt}_j$ is computed according to $\mathsf{PIRSecurity}^1_{\mathsf{PIR},\mathcal{A}}(1^\lambda, 1^N)$, while
- $\mathsf{ctxt}_{j+1}, \ldots, \mathsf{ctxt}_\ell$ is computed according to $\mathsf{PIRSecurity}^0_{\mathsf{PIR},\mathcal{A}}(1^\lambda, 1^N)$.

Clearly, Hyb_0 is identical to $\mathsf{PIRSecurity}^0_{\mathsf{PIR},\mathcal{A}}(1^\lambda, 1^N)$ and Hyb_ℓ is identical to $\mathsf{PIRSecurity}^1_{\mathsf{PIR},\mathcal{A}}(1^\lambda, 1^N)$. It suffices to show that, for any $j \in [\ell]$, any (two-stage) PPT

adversary $\mathcal{A}$, and any $N = \mathsf{poly}(\lambda)$ it holds that

$$\left|\Pr\left[\mathsf{Hyb}_{j-1} = 1\right] - \Pr\left[\mathsf{Hyb}_j = 1\right]\right| \leq \mathsf{negl}(\lambda).$$

To show the above, we note that

$$\begin{aligned} & \left|\Pr\left[\mathsf{Hyb}_{j-1} = 1\right] - \Pr\left[\mathsf{Hyb}_j = 1\right]\right| \\ \leq\ & \left|\Pr\left[\text{IND-CPA}^0_{\mathsf{FHE},\mathcal{A}}(1^\lambda) = 1\right] - \Pr\left[\text{IND-CPA}^1_{\mathsf{FHE},\mathcal{A}}(1^\lambda) = 1\right]\right|. \end{aligned}$$

By the IND-CPA-security of FHE, we have that for any (two-stage) PPT adversary $\mathcal{A}$

$$\left|\Pr\left[\text{IND-CPA}^0_{\mathsf{FHE},\mathcal{A}}(1^\lambda) = 1\right] - \Pr\left[\text{IND-CPA}^1_{\mathsf{FHE},\mathcal{A}}(1^\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda).$$

The claim then follows.

(c) The communication complexity for the generic construction is $|\mathsf{qry}|+|\mathsf{rsp}| = |\mathsf{pk}|+(\ell+1)\cdot|\mathsf{ctxt}| = p(\lambda) + (\ell+1)\cdot\log|\mathcal{X}|\cdot p(\lambda) = \ell\cdot\log|\mathcal{X}|\cdot\mathsf{poly}(\lambda)$.

(i) Plugging in $\ell = 2$ and $|\mathcal{X}| = 2^{\sqrt{N}}$, we have $|\mathsf{qry}| + |\mathsf{rsp}| = \sqrt{N}\cdot\mathsf{poly}(\lambda)$.

(ii) $\mathcal{X} = \{0,1\}$ so $|\mathcal{X}| = 2$. Since $N = \binom{\ell}{k} = \Theta(\ell^k)$, we have $\ell \leq O(N^{1/k})$. Therefore $|\mathsf{qry}| + |\mathsf{rsp}| = N^{1/k}\cdot\mathsf{poly}(\lambda)$.

(d) Without loss of generality, assume $N = 2^\ell$. Let $\mathcal{X} = \{0,1\}$. Define $\mathsf{Encode}(i) = (i_1,\ldots,i_\ell) \in \{0,1\}^\ell$, such that $\sum_{j=1}^{\ell} i_j 2^{j-1}$. That is, the encoding of i is its bit representation. Define $f_D(X_1,\ldots,X_\ell) = \sum_{i\in[N]} d_i\cdot(1-(X_1-i_1))\ldots(1-(X_\ell-i_\ell))$, which is a degree-$\ell$ polynomial. Observe that if the input $(x_1,\ldots,x_\ell)$ to f_D is an encoding of i, then $f_D(x_1,\ldots,x_\ell) = d_i$. Since $\ell = \log N$ and $|\mathcal{X}| = 2$, the communication complexity is $\log N\cdot\mathsf{poly}(\lambda)$.

Question 3 (Oblivious Evaluation of FHE Ciphertexts)**.** In this question, we consider the setting of symmetric-key FHE (i.e. both Enc and Dec take as input the secret key sk, but Eval still takes as input the public key pk), and study an unusual application of FHE where homomorphic evaluations are performed over ciphertexts which are encrypted under potentially "wrong" keys. We want to design a mechanism so that homomorphic evaluation is performed on two ciphertexts only if they are encrypted under the same key (otherwise we risk destroying the content of the ciphertexts).

To give more context, let $\tilde{\Pi}.(\mathsf{KGen},\mathsf{Enc},\mathsf{Dec},\mathsf{Eval})$ be a symmetric-key fully-homomorphic encryption scheme with message space $\mathcal{X}$. Let $\mathsf{ctxt}^\dagger_x \leftarrow \mathsf{Enc}(\mathsf{sk}^\dagger, x)$ be a *fresh* ciphertext of x under a secret key $\mathsf{sk}^\dagger$, and ctxt_m be a ciphertext such that $\mathsf{Dec}(\mathsf{sk},\mathsf{ctxt}_m) = m$ under a (possibly different) secret key sk. Our goal is to design a conditional evaluation mechanism $\mathsf{CondEval}(\mathsf{pk}, g, \mathsf{ctxt}^\dagger, \mathsf{ctxt})$ which, on input

a public-key pk (for secret-key) sk, a function $g(X,M)$, a fresh ciphertext $\mathsf{ctxt}^\dagger_x$, and a ciphertext ctxt_m

computes a ciphertext which decrypts (under sk) to

$$g(x,m) \text{ if } \mathsf{sk} = \mathsf{sk}^\dagger, \text{ and } m \text{ otherwise,}$$

all the while without knowing $\mathsf{pk}^\dagger$ (and obviously not sk nor $\mathsf{sk}^\dagger$). We can think of x as being some instruction for modifying m, which only applies when the keys match, i.e. $\mathsf{sk} = \mathsf{sk}^\dagger$. For the problem to be non-trivial, we need that $\mathsf{ctxt}^\dagger_x$ does not reveal information about $\mathsf{pk}^\dagger$, which we assume to be the case.

Before describing our construction, we introduce the following auxiliary function: Let $\mathsf{VerifyEval}_{\Sigma,\mathsf{pk}_\Sigma,g}$ be a function which is parametrised by a signature scheme Σ, a public key pk_Σ of Σ, and a function $g(X,M)$,

$\tilde{\Pi}.\mathsf{KGen}(1^\lambda)$

$(\mathsf{pk}_\Pi, \mathsf{sk}_\Pi) \leftarrow \Pi.\mathsf{KGen}(1^\lambda)$
$(\mathsf{pk}_\Sigma, \mathsf{sk}_\Sigma) \leftarrow \Sigma.\mathsf{KGen}(1^\lambda)$
$\mathsf{pk} := (\mathsf{pk}_\Pi, \mathsf{pk}_\Sigma)$
$\mathsf{sk} := (\mathsf{sk}_\Pi, \mathsf{sk}_\Sigma)$
return $(\mathsf{pk}, \mathsf{sk})$

$\tilde{\Pi}.\mathsf{Enc}(\mathsf{sk}, x)$

$\sigma \leftarrow \Sigma.\mathsf{Sign}(\mathsf{sk}_\Sigma, x)$
$\mathsf{ctxt}_0 \leftarrow \Pi.\mathsf{Enc}(\mathsf{sk}_\Pi, x)$
$\mathsf{ctxt}_1 \leftarrow \Pi.\mathsf{Enc}(\mathsf{sk}_\Pi, \sigma)$
$\mathsf{ctxt} := (\mathsf{ctxt}_0, \mathsf{ctxt}_1)$
return ctxt

$\tilde{\Pi}.\mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt})$

$x \leftarrow \Pi.\mathsf{Dec}(\mathsf{sk}_\Pi, \mathsf{ctxt}_0)$
return x

$\tilde{\Pi}.\mathsf{Eval}(\mathsf{pk}, f, (\mathsf{ctxt}_i)_{i=1}^{\ell})$

$\mathsf{ctxt}_0 \leftarrow \Pi.\mathsf{Eval}(\mathsf{pk}_\Pi, f, (\mathsf{ctxt}_{1,0}, \ldots, \mathsf{ctxt}_{\ell,0}))$
$\mathsf{ctxt}_1 := \bot$
$\mathsf{ctxt} := (\mathsf{ctxt}_0, \mathsf{ctxt}_1)$
return ctxt

$\tilde{\Pi}.\mathsf{CondEval}(\mathsf{pk}, g, \mathsf{ctxt}^\dagger, \mathsf{ctxt})$

$\mathsf{ctxt}_0 \leftarrow \Pi.\mathsf{Eval}(\mathsf{pk}_\Pi, \mathsf{VerifyEval}_{\Sigma,\mathsf{pk}_\Sigma,g}, (\mathsf{ctxt}_0^\dagger, \mathsf{ctxt}_1^\dagger, \mathsf{ctxt}_0))$
$\mathsf{ctxt}_1 := \bot$
$\mathsf{ctxt} := (\mathsf{ctxt}_0, \mathsf{ctxt}_1)$
return ctxt

Figure 8.1: Construction of symmetric-key FHE Π with conditional evaluation mechanism.

and which computes

$$(x, \sigma, m) \mapsto \begin{cases} g(x, m) & \Sigma.\mathsf{Vf}(\mathsf{pk}_\Sigma, x, \sigma) = 1 \\ m & \Sigma.\mathsf{Vf}(\mathsf{pk}_\Sigma, x, \sigma) = 0, \end{cases}$$

i.e. if Σ accepts the signature σ of x, then $\mathsf{VerifyEval}_{\Sigma,\mathsf{pk}_\Sigma,g}(x, \sigma, m)$ outputs $g(x, m)$, otherwise it outputs m.

In Fig. 8.1, we construct a symmetric-key FHE $\tilde{\Pi}.(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ equipped with an additional algorithm $\mathsf{CondEval}$, from an underlying symmetric-key FHE $\Pi.(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ and a signature scheme $\Sigma.(\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vf})$, all with message space $\mathcal{X}$, using a traditional "sign-then-encrypt" approach.

(a) Assume that the underlying symmetric FHE scheme is perfectly correct (as defined in the lecture notes). Formalise a notion of "conditional evaluation correctness" and provide an explanatory text. Points to consider include but are not limited to:

- What are the quantifiers for ciphertexts to be considered? For example, is it realistic to consider all possible ciphertexts, or only "honestly generated" ones?
- Should the property hold perfectly, statistically, or only against PPT algorithms?

(b) Based on the correctness of the underlying FHE scheme Π and the signature scheme Σ, as well as the unforgeability of Σ under chosen message attacks, give a high-level argument for why FHE scheme Π constructed in Fig. 8.1 satisfies your notion of conditional evaluation correctness defined in Part (a). Revisit your answer in Part (a) if your notion turns out too strong to be satisfiable.

(c) (Optional). Suggest an application of FHE with a conditional evaluation mechanism.

(a) We say that $\tilde{\Pi}$ has conditional evaluation correctness if for any $(x, m) \in \mathcal{X}^2$ and any function $g : \mathcal{X}^2 \to \mathcal{X}$ it holds that

$$\Pr\left[\mathsf{Corr}_{\tilde{\Pi},x,m,g}(1^\lambda) = 0\right] \leq \mathsf{negl}(\lambda),$$

where the correctness experiment $\mathsf{Corr}_{\tilde{\Pi},x,m,g}$ is defined as follows:

$\underline{\mathsf{Corr}_{\bar{\Pi},x,m,g}(1^\lambda)}$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
$(\mathsf{pk}^\dagger,\mathsf{sk}^\dagger) \leftarrow \mathsf{KGen}(1^\lambda)$
$\mathsf{ctxt}_m \leftarrow \mathsf{Enc}(\mathsf{sk},m)$
$\mathsf{ctxt}_x \leftarrow \mathsf{Enc}(\mathsf{sk},x)$
$\mathsf{ctxt}' \leftarrow \mathsf{CondEval}(\mathsf{pk},g,\mathsf{ctxt}_x,\mathsf{ctxt}_m)$
if $\mathsf{Dec}(\mathsf{sk},\mathsf{ctxt}') \neq g(x,m)$ **then return** 0
$\mathsf{ctxt}_x^\dagger \leftarrow \mathsf{Enc}(\mathsf{sk}^\dagger,x)$
$\mathsf{ctxt}'' \leftarrow \mathsf{CondEval}(\mathsf{pk},g,\mathsf{ctxt}_x^\dagger,\mathsf{ctxt}_m)$
if $\mathsf{Dec}(\mathsf{sk},\mathsf{ctxt}'') \neq m$ **then return** 0
return 1

Explanation. The plaintext which is expected to be inside ctxt' is $g(x,m)$, since ctxt_m and ctxt_x are encrypted under the same secret key sk, whereas the plaintext which is expected to be inside ctxt'' is m, since ctxt_m is encrypted under sk but $\mathsf{ctxt}_x^\dagger$ under $\mathsf{sk}^\dagger \neq \mathsf{sk}$ (with overwhelming probability).

Remark. The above definition captures evaluation correctness for honestly generated $\mathsf{ctxt}_x^\dagger$. However, in some settings one may want to protect against an adversary which tries to come up with a malicious $\mathsf{ctxt}_x^\dagger$ which, under conditional evaluation, corrupts the message encrypted in ctxt_m.

(b) The perfect correctness of the FHE scheme Π and the perfect correctness of the signature scheme Σ imply that the probability that

$$\mathsf{Dec}(\mathsf{sk},\mathsf{ctxt}') \neq g(x,m)$$

is 0. To argue that the probability that

$$\mathsf{Dec}(\mathsf{sk},\mathsf{ctxt}'') \neq m$$

is negligible, perhaps surprisingly, we will argue via *reduction* to UNF-CMA-security.

Assume towards contradiction that there exist x,m,g such that

$$\mathsf{Dec}(\mathsf{sk},\mathsf{ctxt}'') \neq m$$

in the correctness game with non-negligible probability. We construct the following PPT adversary $\mathcal{A}_{x,m,g}$ against UNF-CMA security of the signature scheme Σ. The idea is to simply generate a signature $\sigma^\dagger$ of x under an independent signing key $\mathsf{sk}_\Sigma^\dagger$, and submit $(x,\sigma^\dagger)$ as forgery. Concretely, we construct $\mathcal{A}_{x,m,g}$ as follows.

$\underline{\mathcal{A}_{x,m,g}^{\mathsf{GETPK},\mathsf{SIG},\mathsf{VERIFY}}(1^\lambda)}$
$\mathsf{pk}_\Sigma \leftarrow \mathsf{GETPK}$
$(\mathsf{pk}_\Sigma^\dagger,\mathsf{sk}_\Sigma^\dagger) \leftarrow \Sigma.\mathsf{KGen}(1^\lambda)$
$\sigma \leftarrow \Sigma.\mathsf{Sign}(\mathsf{sk}^\dagger,x)$
return $\mathsf{VERIFY}(x,\sigma)$

Clearly, $\mathcal{A}_{x,m,g}$ runs in $\mathsf{poly}(\lambda)$ time. It remains to argue that $\mathcal{A}_{x,m,g}$ has non-negligible advantage against the UNF-CMA-security of Σ.

First, observe that σ generated by $\mathcal{A}_{x,m,g}$ has the same distribution as the signature encrypted in $\mathsf{ctxt}^\dagger_x$ in the correctness experiment $\mathsf{Corr}_{\Pi,x,m,g}$. Since $\mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt}'') \neq m$ with non-negligible probability, by the perfect correctness of the FHE scheme Π, we have $\Sigma.\mathsf{Vf}(\mathsf{pk}_\Sigma, x, \sigma) \neq 0$ with non-negligible probability, which contradicts the UNF-CMA-security of Σ.

Remark. The above proof still works even when considering conditional evaluation correctness against malicious $\mathsf{ctxt}^\dagger_x$.

(c) One possible application could be an "oblivious" firewall which should allow ciphertexts encrypted under sk_1,..,sk_ℓ to pass through, but not ciphertexts under any other $\mathsf{sk}^\dagger \notin \{\mathsf{sk}_1, .., \mathsf{sk}_\ell\}$. The firewall is untrusted in the sense that it should not know any of the $\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell$.

The firewall holds ciphertexts $\mathsf{ctxt}_1, \ldots, \mathsf{ctxt}_\ell$, which are all encryptions of some fixed $m^* \in \mathcal{X}$ (e.g. the all-zero string) under $\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell$ respectively. Let $g(X, M) = X$.

On input $\mathsf{ctxt}^\dagger$ supposedly encrypting some message x under $\mathsf{sk}^\dagger$, the firewall then runs the following code

for $i = 1, \ldots, \ell$
  $\mathsf{ctxt}'_i \leftarrow \mathsf{CondEval}(\mathsf{pk}_i, g, \mathsf{ctxt}^\dagger, \mathsf{ctxt}_i)$
return $(\mathsf{ctxt}'_1, \ldots, \mathsf{ctxt}'_\ell)$

The i-th $\mathsf{CondEval}$ evaluation with $\mathsf{sk}^\dagger \neq \mathsf{sk}_i$ results in a ciphertext encrypting a constant message m^* which contains no information about x, whereas if $\mathsf{sk}^\dagger = \mathsf{sk}_i$ then the result would be a ciphertext encrypting x under sk_i.