Advanced Topics on Privacy Enhancing Technologies
CS523
Homomorphic Encryption Exercises

1 RSA

Show that RSA is multiplicatively homomorphic, i.e., $RSA(m) \times RSA(m')$ mod N $= RSA(m \cdot m')$.

Solution:

$$\begin{aligned} RSA(m) \times RSA(m') \text{ mod N} &= (m^e \text{ mod N}) \times (m'^e \text{ mod N}) \text{ mod N} \\ &= (m^e \times m'^e) \text{ mod N} \\ &= (m \times m')^e \text{ mod N} \\ &= RSA(m \times m') \end{aligned}$$

2 Circuit depth

The notion of circuit depth is very important when making use of a leveled HE scheme. This is because this type of scheme only supports a limited number of successive multiplications (its number of "levels") before having to decrypt the result.

We define the depth of a circuit as the minimum of consecutive multiplications required to evaluate a circuit. For example, the circuit $(a \cdot b) + (c \cdot d)$ has a depth of one because both multiplications can be carried out in parallel, they do not depend on each other inputs or outputs. However the circuit $(a \cdot b) \cdot (c + d)$ has a depth of two because the second multiplication takes as input the result of the first multiplication.

Being able to analyze and optimize the depth of a circuit is, therefore, a central task when making use of a leveled HE scheme as it will allow to optimize its parameterization and efficiency.

What is the minimum multiplicative depth of the following circuits ?

1. $f(x, y) = (a \cdot x) \cdot (b \cdot y)$ with $a, b \neq 0$.

2. $f(x) = x^{1024}$

3. $f(x) = a + b \cdot x + c \cdot x^3 + d \cdot x^5$ with $a, b, c, d \neq 0$.

4. $f(x) = \sum_{i=0}^{\ell-1} a_i \cdot x^i$ with $a_i \neq 0$

5. $f(x_{0 \leq i < n}) = h(g(h(x_{0 \leq i < n})))$ given that $h(x_{0 \leq i < n}) = y_{0 \leq i < n}$ where $y_i = \sum_{j=0}^{n-1} a_{i,j} \cdot x_j$ with $a_{i,j} \neq 0$, and that $g(x_{0 \leq i < n}) = y_{0 \leq i < n}$ where $y_i = \sum_{j=0}^{\ell-1} b_j \cdot x_i^j$ with $b_j \neq 0$.

Solution:

1. 2

2. $\lceil \log_2(1024) \rceil = 10$

3. $\lceil \log_2(5) \rceil + 1 = 4$ (but it is in practice possible to do it in 3)

4. $\lceil \log_2(\ell - 1) \rceil + 1$ (but it is in practice possible to do it in $\lceil \log_2(\ell - 1) \rceil$)

5. $1 + (\lceil \log_2(\ell - 1) \rceil + 1) + 1$ (but it is in practice possible to do it in $\lceil \log_2(\ell - 1) \rceil + 2$)

3 Evaluating functions

Most of the time a HE scheme can only evaluate a few basis operations like additions and multiplications. Those can, however, be used as building blocks to construct more complicated and more useful functions. In this exercise, we will see how to use simple operations to evaluate complicated functions, and how some operations, which could be thought as simple at first glance, are in fact complicated to evaluate when given only a limited number of basic operations.

Assume that you are given a HE scheme that can encrypt vectors of n floating points numbers of the form $a = (a_0, \ldots, a_{n-1})$ and evaluate on them three operations Add, Mul, Rotate which are defined as

$$\begin{aligned} \mathsf{Add}(a, b) &: a_i + b_i \text{ for } 0 \leq i < n, \\ \mathsf{Mul}(a, b) &: a_i \cdot b_i \text{ for } 0 \leq i < n, \\ \mathsf{Rotate}(a, k) &: a_{i-k \pmod n} \text{ for } 0 \leq i < n. \end{aligned}$$

Using those three basis operations, explain how you would evaluate the following circuits (you can assume that the scheme supports an unlimited number of operations and that it is also possible to add and multiply by plaintext vectors):

1. $\mathsf{avg}(a) : (\frac{1}{n} \sum_{i=0}^{n-1} a_i, \ldots, \frac{1}{n} \sum_{i=0}^{n-1} a_i)$

2. $\mathsf{exp}(a) : (e^{a_0}, \ldots, e^{a_{n-1}})$
3. $M \cdot a$ where M is an n by n matrix
4. $\mathsf{abs}(a) : (|a_0|, \ldots, |a_{n-1}|)$
5. $\mathsf{inv(a)} : (\frac{1}{a_0}, \ldots, \frac{1}{a_{n-1}})$
6. $\mathsf{max}(a) : (a_i, \ldots, a_i)$ where a_i is the maximum value of a
7. $\mathsf{floor(a)} : (\lfloor a_0 \rfloor, \ldots, \lfloor a_{n-1} \rfloor)$

Solution:

1. We need $\log_2(n)$ rotations and additions to do the inner sum and one scalar multiplication by $1/n$.
2. We can approximate e with a polynomial and evaluate that polynomial, which can be done solely with additions, non-scalar and scalar multiplications.
3. (a) Naive : store each row of the matrix (so n rows), multiply each on of them with the ciphertext (so n multiplications) and do an inner sum using rotations and additions (so $n \log_2(n)$ rotations and additions). Then repack the n ciphertexts into one ciphertext (so $n-1$ rotations and n additions.
   (b) Better : store the matrix in its diagonal form and use a baby-step giant-step algorithm to evaluate it, uses $O(\sqrt{n})$ rotations and $O(n)$ multiplications and additions.
4. (a) Compute the square and then approximate the square root with a polynomial.
   (b) Approximate directly the absolute value within the desired range using a polynomial.
5. (a) Divide the value by a constant to make them fall within a specific interval, then use an iterative algorithm using additions and multiplications to compute the inverse, and divide the result by the initial scaling constant to obtain the correct result (https://eprint.iacr.org/2016/421.pdf Section 4.2).
   (b) Approximate directly the inverse within a specific range of interest using a polynomial.
6. (a) We can use the formula $\max(a, b) = (a+b+|a-b|)/2$, approximating the absolute value as explain in 4), then compare pairs of elements $\sum_{i=1}^{\lceil \log_2(n) \rceil} n \cdot 2^{-i}$ times to obtain the max of the vector.
   (b) We can use the following iterative algorithm :
      i. Choose the number of iterations r (2 is usually sufficient) and set $\omega = x$.

ii. For r iterations, compute $\omega = e^{\omega_{0 \le i < n}}$, then divide ω by its average.

iii. Compute $\max(x) = \frac{1}{n} \sum x_i \cdot \omega_i$.

7. If $|x_i - \lfloor x_i \rfloor|$ is small, then we have $\lfloor x_i \rfloor \approx x_i - \frac{1}{2\pi} \sin(2\pi x_i)$, which can be approximated by a polynomial. If $|x_i - \lfloor x_i \rfloor|$ is not small, then there is no know way to do it efficiently.

References

[1] F. McSherry, "Privacy integrated queries:an extensible platform for privacy-preserving data analysis," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data (SIGMOD)*. Association for Computing Machinery, Inc., June 2009, for more information, visit the project page: http://research.microsoft.com/PINQ. [Online]. Available: https://www.microsoft.com/en-us/research/publication/privacy-integrated-queries/