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 $= RSA(m \cdot m')$.

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

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. $\mathsf{floor(a)} : (\lfloor a_0 \rfloor, \ldots, \lfloor a_{n-1} \rfloor)$

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/