Homomorphic Encryption Primer

Justin Wyss-Gallifent

August 18, 2021

1 Introduction . . . 2
2 Primary Abstract Goal . . . 2
3 Simple Obfuscation of Single Operation . . . 3
4 Ring-Related Items . . . 4
4.1 Rings . . . 4
4.2 Ring Homomorphisms . . . 4
4.3 Ring Isomorphisms . . . 5
5 Noise . . . 6
6 RSA and ElGamal and 7
7 Lattice-Based Cryptography . . . 7
8 History and the Current State: . . . 7

1 Introduction

The underlying drive for homomorphic encryption is the idea of being able to process encrypted data to get encrypted results without ever being able to understand what the data is. While this concept may seem bizarre the notion of working on values as a proxy for other values is fairly simple to implement, as we'll see, it's the desire for asymmetric encryption that complicates things.

2 Primary Abstract Goal

The primary goal is for Bob to select two numbers A and B encrypt them, send them to Alice, have Alice do some computation and return the encrypted values which represent $A + B$ and AB. At no point should Alice be able to know what A and B are. Once Alice can do this then arbitrary complex calculations can be done.

We will not accomplish this with full asymmetric encyption because it is beyond the scope of our shared mathematical knowledge but we will look into some of the underlying ideas.

3 Simple Obfuscation of Single Operation

Here's an example of obfuscating one operation whereby Bob can have Alice do a calculation without obviously knowing what the calculation is.

Example 3.1. Suppose Bob wants Alice to calculate AB where $A, B \in \mathbb{Z}^+$. He creates the encryption function:

$$\mathcal{E}(x) = \lg x$$

He then calculates $\mathcal{E}(A) = \lg A$ and $\mathcal{E}(B) = \lg B$ and sends those to Alice and tells her to add them and send back the result. Alice calulates $\lg A + \lg B$ which she returns to Bob. Bob then decrypts via

$$\mathcal{D}(x) = 2^x$$

Because:

$$\mathcal{D}(\lg A + \lg B) = 2^{\lg A + \lg B} = AB$$

Provided that Alice doesn't know $\mathcal{E}$ or $\mathcal{D}$ then Bob is safe.
Notice in this example that in general we have:

$$\mathcal{D}(\mathcal{E}(x) + \mathcal{E}(y)) = 2^{\lg x + \lg x} = 2^{\lg x} 2^{\lg y} = xy$$

Example 3.2. Note that if Bob wanted Alice only to do multiplication that might be fine and on the other hand if Bob wanted Alice only to do addition he could use $\mathcal{E}(x) = 2^x$ instead, telling her to multiply the encyrpted values, and then $\mathcal{D}(x) = \lg x$. In this case we'd have:

$$\mathcal{D}(\mathcal{E}(x)\mathcal{E}(y)) = \lg(2^x 2^y) = \lg(2^x) + \lg(2^y) = x + y$$

A question and a note:

1. Could we find an encryption function which would enable both operations to be done?
2. One thing that makes these examples confusing is that in the first one, if Bob wants Alice to multiply, he tells her to add, and in the second one, if Bob wants Alice to add, he tells her to multiply. Ideally we'd like:

$$\mathcal{D}(\mathcal{E}(x) + \mathcal{E}(y)) = x + y \text{ and } \mathcal{D}(\mathcal{E}(x)\mathcal{E}(y)) = xy$$

4 Ring-Related Items

4.1 Rings

Loosely speaking a ring is a collection of values (or elements) in which we have both addition and multiplication.

Example 4.1. The set of integers $\mathbb{Z}$ is a ring.

Example 4.2. The set of even integers $2\mathbb{Z}$ is a ring.

Example 4.3. The set $\{0, 1, 2, 3, 4, 5, 6\}$ of integers with all operations taken mod 7 is a ring, denoted $\mathbb{Z}_7$.

Example 4.4. The set of 2×2 matrices with real entries is a ring, denoted $M_2\mathbb{R}$.

Example 4.5. Another example is $\mathbb{Z}_3 \times \mathbb{Z}_2$ which consists of the pairs (a, b) with $a \in \mathbb{Z}_3$ and $b \in \mathbb{Z}_2$ and operations done with corresponding elements, so $(a, b) + (c, d) = (a + c, b + d)$ and $(a, b)(c, d) = (ac, bd)$.

4.2 Ring Homomorphisms

Given two rings R and S, a ring homomorphism is a mapping $\phi : R \to S$ having the properties that $\phi(x + y) = \phi(x) + \phi(y)$ and $\phi(xy) = \phi(x)\phi(y)$.

Example 4.6. The mapping $\phi : \mathbb{Z} \to \mathbb{Z}_{10}$ given by $\phi(x) = x \mod 10$ is a ring homomorphism because:

$$\begin{aligned} \phi(x + y) &= (x + y) \mod 10 \\ \phi(x) + \phi(y) &= (x \mod 10) + (y \mod 10) = (x + y) \mod 10 \end{aligned}$$

and

$$\begin{aligned} \phi(xy) &= (xy) \mod 10 \\ \phi(x)\phi(y) &= (x \mod 10)(y \mod 10) = (xy) \mod 10 \end{aligned}$$

Observe that ϕ is onto but not 1-1.

Example 4.7. The mapping $\phi : \mathbb{Z} \to 2\mathbb{Z}$ given by $\phi(x) = 2x$ is not a ring homomorphism because:

$$\phi(xy) = 2(xy) \text{ but } \phi(x)\phi(y) = (2x)(2y) = 4xy$$

4.3 Ring Isomorphisms

A ring isomorphism is a ring homomorphism ϕ which has an inverse ϕ^{-1}.

Example 4.8. The mapping $\phi : \mathbb{Z}_6 \to \mathbb{Z}_2 \times \mathbb{Z}_3$ given as follows is a ring isomorphism with inverse:

$$\begin{aligned} \phi(0) &= (0,0) & \phi^{-1}(0,0) &= 0 \\ \phi(1) &= (1,1) & \phi^{-1}(1,1) &= 1 \\ \phi(2) &= (0,2) & \phi^{-1}(0,2) &= 2 \\ \phi(3) &= (1,0) & \phi^{-1}(1,0) &= 3 \\ \phi(4) &= (0,1) & \phi^{-1}(0,1) &= 4 \\ \phi(5) &= (1,2) & \phi^{-1}(1,2) &= 5 \end{aligned}$$

It also satisfies the homomorphism properties. This is not obvious but we can check it case-by-case on all 36 combinations. For example here is one check:

$$\begin{aligned} \phi(2 \cdot 5) = \phi(10) = \phi(4) = (0,1) \\ \phi(2)\phi(5) = (0,2)(1,2) = (0,4) = (0,1) \end{aligned}$$

And here is another:

$$\begin{aligned} \phi(2+5) = \phi(7) = \phi(1) = (1,1) \\ \phi(2) + \phi(5) = (0,2) + (1,2) = (1,4) = (1,1) \end{aligned}$$

Important: Note that in such a scenario if we let $\mathcal{E} = \phi$ and $\mathcal{D} = \phi^{-1}$ then we have:

$$\mathcal{D}(\mathcal{E}(x) + \mathcal{E}(y)) = \phi^{-1}(\phi(x) + \phi(y)) = \phi^{-1}(\phi(x+y)) = x + y$$

And:

$$\mathcal{D}(\mathcal{E}(x)\mathcal{E}(y)) = \phi^{-1}(\phi(x)\phi(y)) = \phi^{-1}(\phi(xy)) = xy$$

This is exactly what we wanted, ideally!

Example 4.9. It follows that if Bob were only working with $\mathbb{Z}_6$ he could use $\mathcal{E} = \phi$ as his encryption and $\mathcal{D} = \phi^{-1}$ as his decryption and have Alice do the same operations he wants done, assuming she can't decypher that for example the $(1,0)$ she is working with means 3 to Bob.
This means, for example, that if Bob wants Alice to calculate $4(2+3)$ he calculates $\phi(4) = (0,1)$, $\phi(2) = (0,2)$, and $\phi(3) = (1,0)$. He sends her the set $\{(0,1),(0,2),(1,0)\}$ with instructions such as "Add the last two and multiply by the first". She does $(0,1)((0,2)+(1,0)) = (0,1)(1,2) = (0,2)$ and sends it back. Bob does $\mathcal{D}(0,2) = 2$ so he knows that $4(2+3) = 2$ and Alice did the work for him.

5 Noise

Returning to our opening example of $\mathcal{E}(x) = \lg x$ it's not actually feasible for Bob to send $\lg x$ to Alice because $\lg x$ will probably have infinitely many digits. He can't actually send a message like "lg 3" because that's a give-away. Instead therefore he calculates some digits of $\lg x$ and sends those. The result from Alice will be noisy but could be good enough.

Example 5.1. Bob wants Alice to calculate the product $3 \cdot 5$. He calculates $\lg 3 = 1.584962500721156...$ and $\lg 5 = 2.321928094887362...$ and decides to send her 1.58 and 2.32. She calculates $1.58 + 2.32 = 3.90$ and sends it back. Bob then calculates $2^{3.90} = 14.928527864588919...$ and deduces that the answer is 15.

This will only work to a certain degree. If he had sent her 1.5 and 2.3 then when she returned 3.8 he would find $2^{3.8} = 13.928809012737984...$ and might deduce that the answer is 14, which it isn't.

So perhaps he believes that using two digits beyond the decimal point is sufficient. But it's not.

Example 5.2. Let's be official and define $R(x, n)$ to be x rounded to the closest decimal with n digits beyond the decimal, so for example $R(1.584, 2) = 1.58$ and $R(1.588, 2) = 1.59$. When we're right in the middle we'll round up, so $R(1.585, 2) = 1.59$ and $R(1.5, 0) = 2$. Then we put:

$$\mathcal{E}(x) = R(\lg x, 2)$$
$$\mathcal{D}(x) = R(2^x, 0)$$

Suppose Bob wants Alice to calculate $3^2 \cdot 5 \cdot 7 \cdot$. He calculates his two-digit approximations:

$$\mathcal{E}(3) = R(\lg 3, 2) = 1.58$$
$$\mathcal{E}(5) = R(\lg 5, 2) = 2.32$$
$$\mathcal{E}(7) = R(\lg 7, 2) = 2.81$$

He then sends $(1.58, 1.58, 2.32, 2.81)$ to Alice with instructions to add. She does so and gets 8.29 which she sends back to Bob. Bob then calculates:

$$\mathcal{D}(2^{8.29}) = R(2^{8.29}, 0) = 313$$

This is incorrect. The correct answer is $3^2 \cdot 5 \cdot 7 = 315$.

What we're seeing here is that as the calculations get more complex the noise accumulates. Broadly speaking there are a few ways to adjust for this:

1. Bob could only send simple calculations.
2. Bob could pre-emptively determine how bad the noise could be and adjust what he sends accordingly.
3. Bob could give Alice some obscure instructions which would allow her to correct for the noise without realizing that she is correcting for the noise.

6 RSA and ElGamal and ...

Some things we know:

Example 6.1. RSA is known as partially homomorphic, meaing it works for one operation.

Observe that RSA works for multiplication because:

$$\mathcal{E}(x) \equiv x^e \mod n$$

Thus:

$$\mathcal{E}(xy) \equiv (xy)^e \equiv x^e y^e \equiv \mathcal{E}(x)\mathcal{E}(y) \mod n$$

Example 6.2. The shift cipher:

$$\mathcal{E}(x) \equiv x + b \mod 26$$

is not even partially homomorphic. It does not work with either addition or multiplication.

7 Lattice-Based Cryptography

Lattice-based cryptography was developed in the mid 2000s and opened up the possibility. A lattice is, very loosely speaking, a grid of points in $\mathbb{R}^n$. For example the set of integer points in the plane is a lattice, as is the set of integers points $\{(2x, x + 3y) \,|\, x, y \in \mathbb{Z}\}$. There are many hard lattice-based problems which may be used to develop encryption schemes.

One is the shortest vector problem (SVP) which asks us to find the shortest nonzero vector in a given lattice. A lattice can be given in terms of a bais and if the basis vectors are long then figuring out how to combine them to form a short vector is difficult. This problem gets even harder when n increases and lattice-based cyptography usually works in dimensions upwards of 10000.

Lattice-based cryptography is understood to be more secure than RSA or ElGamal, for example, because while we know that factoring and discrete logarithms can be solved in polynomial time on a quantum computer it is believed that lattice problems cannot be.

Many lattice-based schemes involve adding some noise to the message. If the noise level is low then as above we can correct for it. Unfortunately, also as above, as the calculations get more complicated the noise grows so large that we cannot correct for it.

8 History and the Current State:

Up until the early 2000s it was not clear that asymmetric homomorphic encryption could even exist.

Lattice-based crypography was developed in the mid 2000s.

Asymmetric fully homomorphic encryption based upon Lattices was first developed in 2009 by Craig Gentry based upon lattice-based cryptography.

Lattice-based cryptography systems generally introduce noise into the ciphertext. Therefore various methods must be implemented to deal with this noise.

This is where we are today, in 2021.