HOMOMORPHIC ENCRYPTION

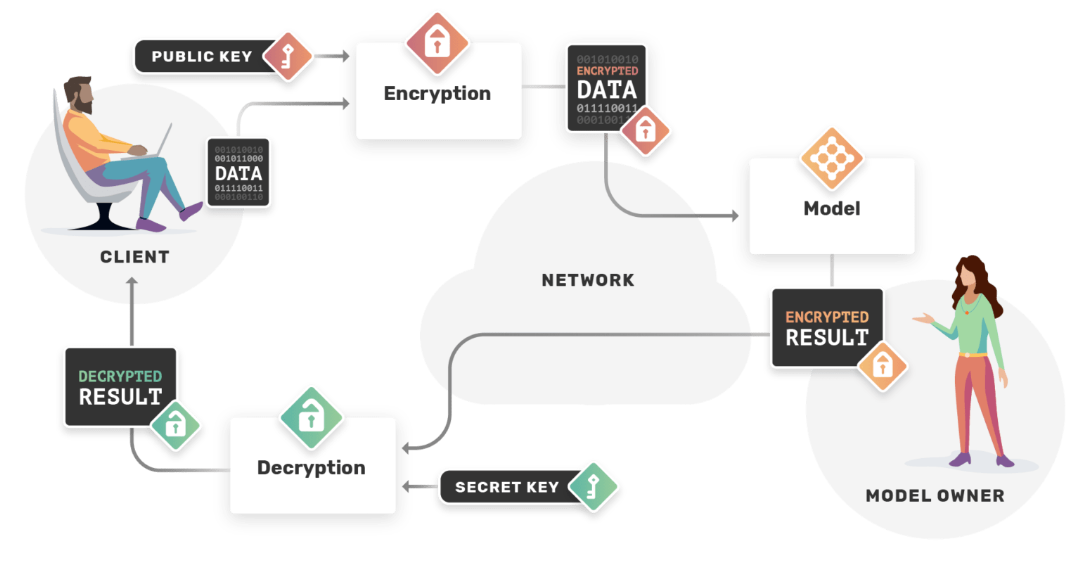
1. An introduction to the cryptographic approach.

The word Homomorphic means a structure that preserve the map between two similar algebraic function, this structure can be two groups or rings or vector spaces.

Homomorphic encryption is a type of encryption that allows user to perform binary encryption without decrypting the data. So basically we can use and process the encrypted data and give access to third parties without compromising the confidentiality of the data. Homomorphic encryption is highly privacy friendly approach to maintain the confidentially of a data while utilizing it for various important tasks.

For example if we talk about the Healthcare data that is a PHI standard, Protected Health Information states that the data regarding to the health of patients is highly sensitive and the access should not be given to anyone. So generally we use the encryption and we encrypt the data with a key and then make it secure and confidential but when we need to process that data for some statistical analysis or some other research we first decrypt it and then perform some other important activities. This approach is neither an ideal approach nor a secure approach. As it take time to decrypt and encrypt the data and then privacy of data is subjected to the use as it might not be used properly and get compromised. So the Homomorphic encryption is a solution to this problem. We can analyse and process the data without decrypting it so without compromising the secrecy.

1. Technical details into how and why the approach works.



(Antonio Lopardo-2019)

Homomorphic encryption has three types.

**Partially Homomorphic Encryption (PHE)**

The partially Homomorphic encryption allows to perform only one operation on the cipher text for unlimited times.

The operation supported in PHE is only can be addition or multiplication.

**Some What Homomorphic encryption (SHE)**

This Homomorphic encryption allows to perform both addition and multiplication on the cipher text but for the limited times.

**Fully Homomorphic Encryption (FHE).**

This encryption allows to perform both multiplication and addition on the cipher text for unlimited times. FHE also allows the arbitrary computation on the encrypted data.

**Paillier Cryptosystem**

Paillier Cryptosystem was introduced in 1999 and is based on the Partially Homomorphic Encryption. It is additive Homomorphic type of encryption and allows the addition of two cipher texts but not the multiplication between them.

**Mathematical Description.**

Key Generation.

First thing, we have to choose two prime numbers. These should be random and independent to each other.

These numbers should this property.

gcd(pq,(p-1)(q-1))=1

Calculate.

n= pq, λ = lcm(p-1,q-1)

Lcm= Least Common Multiple

Select a random integer g where g belongs to integers with base of squares.

Make sure that n divides the order of g by checking the existence of the modular multiplicative inverse.

Μ= (L(gλ(mod n²)))-1(mod n)

Where L defined as L(u)= (u-1)/n

Note that a/b does not denotes the modular multiplication of a times the modular multiplicative inverse of b, but rather it shows the quotient of a divided by b. So the public encryption key pair is (n,g) and the private decryption key pair (λ,u)

In case if the p and q are equal in length then the above key step of key generation would be more simpler

g= n+1, λ=φ(n), u= φ(n) ^-1 (mod n)

φ(n) = (p-1)(q-1)

1. An example of encryption and decryption of a sample message.

Encryption.

Let m is the message, a plain text that has to be encrypted.

Where m ɛ Z, Integers.

To compute the cipher text

C = g^m . r^n mod (n²)

 {\displaystyle {\mathcal {E}}(m)=g^{m}r^{n}\;{\bmod {\;}}n^{2}}

Lets p = 7, q = 11.

Then

n = pq = 7x11 = 77.

Select an integer, such that the order of g is multiple of n^2 in Z n^2.

So we randomly choose the integer,

g = 5652

As all conditions are met so we have the public key pair,

(n , g) = (77 , 5652)

To encrypt the message we take the plain text m,

m = 42,

m ɛ Z.

r = 23

Now we compute

C = g^m . r^n mod (n²)

C = (5652)^42. 23^77 mod(77^2)

C = 4624. mod 5929

Decryption.

To decrypt the encrypted message, we needs.

m = L(c^λ(mod n²)). u(mod n)

To decrypt the cipher text,

λ= lcm (6,11)

L(u)= (u-1)/n, compute,

K = L(g^λ mod n^2))

K = L(5652^30 (mod5929)

K= L (3928)

K= L (3928-1)/77

K = 51

Now we have to compute the inverse of k,

u = k ^-1 (mod n)

u = 51^-1

u = 74 mod 77

m = L(c^λ mod n^2).u(mod n)

= L(4624^30. mod 5929). 74mod77

=L(4852). 74mod 77

m = 42

1. A comparison of the approach to other similar techniques, for example those discussed in the course.

Homomorphic encryption is a different approach if compare to other crypto algorithms such as DES and AES because it is an non-symmetric approach but some of its features are similar to the RSA. We have to calculate the key pair via mathematical functions. Homomorphic encryption enables to perform some tasks and functions with the help of algebraic systems. It has some limitations that it requires a lot of resources to perform the encryption, decryption and the processing.

Homomorphic encryption has the following benefits.

* It allows secure and efficient use of data in cloud.
* It enables collaborations between organizations to use the sensitive data.
* It mitigate the risk of exposure hence ensure the regulatory compliances.

References.

* Yi, X., Paulet, R., Bertino, E. (2014). Homomorphic Encryption. In: Homomorphic Encryption and Applications. SpringerBriefs in Computer Science. Springer, Cham. <https://doi.org/10.1007/978-3-319-12229-8_2>
* Monique Ogburn, Claude Turner, Pushkar Dahal, Homomorphic Encryption, Procedia Computer Science, Volume 20, 2013, Pages 502-509, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2013.09.310.>
* Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. 2018. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. ACM Comput. Surv. 51, 4, Article 79 (July 2019), 35 pages. <https://doi.org/10.1145/3214303>
* Paulo Martins, Leonel Sousa, and Artur Mariano. 2017. A Survey on Fully Homomorphic Encryption: An Engineering Perspective. ACM Comput. Surv. 50, 6, Article 83 (November 2018), 33 pages. https://doi.org/10.1145/3124441