Homomorphic Encryption

Prajna Puranik

What is homomorphic encryption?

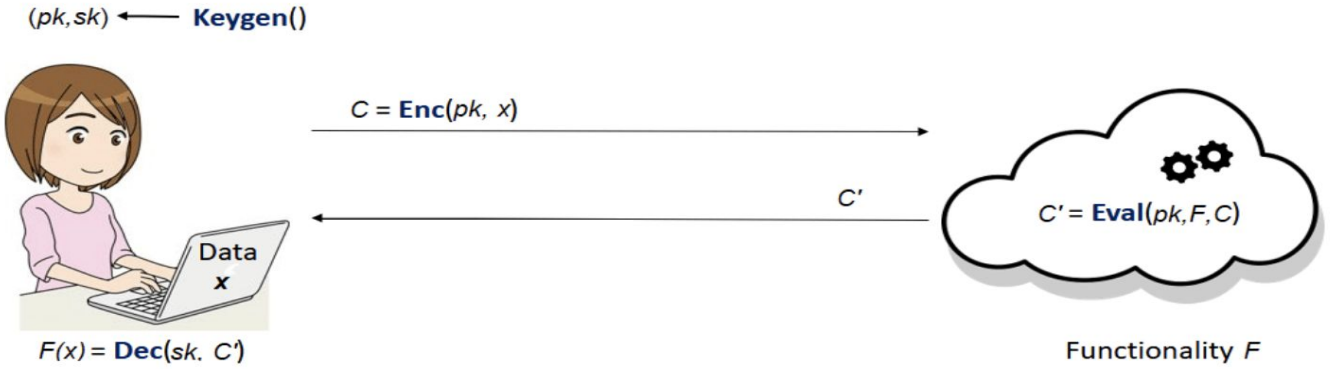
- Homomorphic encryption → Permits users to perform computations on encrypted data without first decrypting it.
- These result is in an encrypted form
- When it is decrypted, result is identical to that produced if the operations been performed on the unencrypted data.
- Homomorphic encryption makes it possible to analyze or manipulate encrypted data without revealing the data to anyone.

Example

- Alice holds some personal information x (e.g. her medical records and her family's medical history).
- There is also a company that makes very good predictions based on this kind of information, expressed as the functionality F
- Alice is very interested in these predictions but is also reluctant to trust the company with her sensitive information.
- The company can't just give their model to Alice to make the predictions herself.
- A solution is to use Homomorphic Encryption

To note:

- Alice sends her data encrypted, so the company never learns anything about x
- Computing on the encrypted data C does not involve Alice's private key. Only her public key pk is used.
- Evaluation algorithm in the company side uses the description of F to do computations on C to get C'
- By using her secret key, Alice manages to recover the information that interests her, namely F(x).



Notes

- Just like other forms of encryption, homomorphic encryption uses a public key to encrypt the data.
- Unlike other forms of encryption, it allows functions to be performed on the data while it's still encrypted.
- Then, the individual with the matching private key can access the unencrypted data after the functions and manipulation are complete.
- This allows the data to be and remain secure and private even when someone is using it.
- Homomorphic encryption has huge potential in areas with sensitive personal data such as in financial services or healthcare when the privacy of a person is paramount.
- Another bonus of homomorphic encryption → unlike other encryption models in use today, it is safe from getting broken by quantum computers.

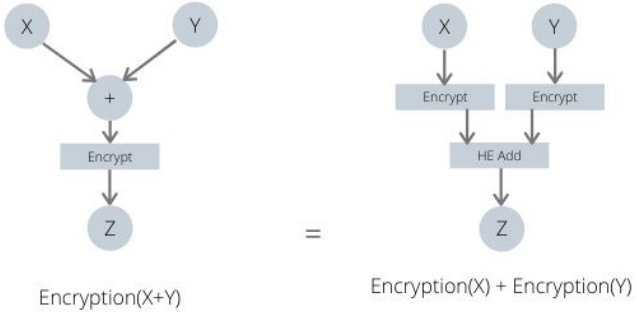
Working

Say we have a function F that performs a computation on two elements x and y and outputs a result z = F(x, y)

An HE scheme lets you perform a certain function F on encrypted elements such that:

F(encrypted(X), encrypted(Y)) = encrypted(F(X, Y))

That function F is generally an addition or a multiplication, having a scheme that supports arbitrary function F is not yet a reality



Types of homomorphic encryption

There are three main types of homomorphic encryption:

- Fully-HE (FHE): keeps information secure and accessible while allowing any number of addition and multiplication operations.
- Somewhat-HE (SHE): It allows both addition and multiplication, but we are limited in term of the number of operations we can perform.
- Partially-HE (PHE):
    a. Keeps sensitive data secure by only allowing select mathematical functions to be performed on encrypted data
    b. This type of scheme either allows addition or multiplication, but in an unlimited fashion.

Why is it not used wide-scale yet?

- The biggest barrier to wide-scale adoption of homomorphic encryption is that it is still very slow—so slow it's not yet practical to use for many applications.
- However, there are companies are working on speeding up the process by decreasing the computational overhead that's required for homomorphic encryption.

References

[1] https://www.freecodecamp.org/news/introduction-to-homomorphic-encryption/

[2] M. O'Keeffe, "The Paillier Cryptosystem A Look Into The Cryptosystem And Its Potential Application", College of New Jersey, 2008

[3] A. Acar et al, "A Survey on Homomorphic Encryption Schemes: Theory and Implementation", Florida International University

Thank you!