Application analysis of data encryption technology

Bingcheng Li[1,4,5], Dingkang Li[2], Mingyuan Zhu[3]

[1]St.Johnsbury Academy Jeju, Jeju 63644, Korea
[2]Vanke Meisha Academy, ShenZhen 518000, China
[3]Nanjing Foreign Language School British Columbia Academy, NanJing 210008, China

[4]s17012424@sjajeju.kr
[5]Corresponding author

Abstract. Data encryption technology is a key method to secure sensitive data. The privacy and confidentiality of data is protected by encrypting the data, i.e., transforming the original data into a form that cannot be understood without authorization. However, the security of traditional encryption algorithms is gradually challenged with the increase in computing power. Therefore, studying how to apply more advanced encryption techniques has become a current research hotspot. This study aims to analyze the application of data encryption techniques and explore the advantages and applicability of emerging encryption algorithms. This paper presents a variety of methods for preventing users' privacy breaches, with a primary focus on the principles of homomorphic encryption and how data can be accessed by users without decryption. Additionally, the working principles of secure multiparty computation are discussed, allowing multiple users to perform calculations on shared data while preserving data privacy. Furthermore, the paper explores data encryption techniques that employ specific algorithms to convert plaintext into ciphertext, ensuring both data consistency and privacy. Finally, a summary and future prospects are provided.

Keywords: Big data, Data encryption technology, Types.

1. Introduction

In the situation of our society right now, our personal data is constantly being collected, transmitted, and analyzed. Whatever it’s stored online or in cloud storage, or business transactions, medical records, our sensitive information is becoming increasingly susceptible to threats from malicious actors and abusers. In that situation, protecting users' data comes first. This is extremely significant so that all users' information is not getting leaked, because once the information is being leaked, then the most obvious consequence is when personal information is being leaked, attackers may use that information to impersonate the victim and engage in identity theft activities. This may include opening credit cards, applying for loans, opening bank accounts for scamming, even threatening in person because hackers know the address once they have got the personal information. If the leaked data includes financial information, victims may suffer economic losses, including bank account theft or misuse of credit cards. There are many methods that have been created or used to prevent those unfortunate things from happening to society. This essay will introduce some methods that can be used to prevent users’ privacy.

mainly focusing on how Homomorphic encryption functions, how the data become Homomorphic and how it allows users to access the data without decrypting it. Also introducing the Secure Multiparty Compute of how it allows different users to use the same data to calculate but also keep the privacy of the data. Lastly, how the Data encryption technology uses specific algorithms to change the plaintext into ciphertext to keep the consistency and privacy.

2. Homomorphic encryption (HE)

Homomorphic encryption has the unique way of protecting data, and is so beneficial for users' privacy. It allows data to be computed while in an encrypted state, without the need for prior decryption. The value of this feature is so beneficial and so valuable for data privacy. When we encrypt our data, even if the service providers or other data processors can access it, they cannot comprehend its content. This means that the risk of leakage of personal sensitive data is significantly reduced, while also providing more control to data owners. The paper [1] mentions that HE is an encryption scheme in cryptography that allows third parties, such as cloud or service providers, to perform computations on encrypted data while preserving the data's structure and format. There is an example picture that explains how HE functions, where users can perform operations on encrypted data without knowing the plaintext values. Figure 1 is an example of how basic Homomorphic encryption processes what has been done in each step.

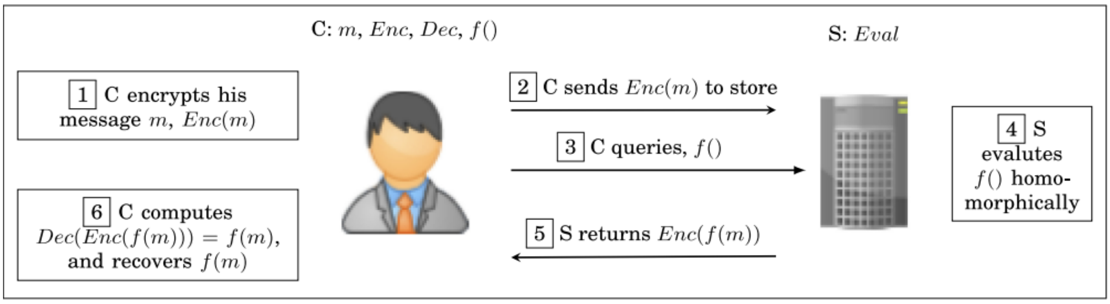
C: m, Enc, Dec, $f()$
S: $Eval$
1 C encrypts his message m, $Enc(m)$
2 C sends $Enc(m)$ to store
3 C queries, $f()$
4 S evalutes $f()$ homo-morphically
5 S returns $Enc(f(m))$
6 C computes $Dec(Enc(f(m))) = f(m)$, and recovers $f(m)$

Figure 1. An example of how basic Homomorphic encryption processes [1]

2.1. PHE (Partially Homomorphic Encryption)

Several different types of encryption exist in Homomorphic encryption. One is Partially Homomorphic Encryption (PHE) which is also able to decrypt data. However, unlike Fully HE, PHE only supports specific types of mathematical calculation, typically including Addition or Multiplication, but not both simultaneously. This makes PHE so useful in certain applications because it allows specific computations to be performed without revealing the original data. There is a brief description and examples of different types of PHE in paper [2].

Let E(K,x) represent the encryption of x with the encryption equation E and the key K, and F represent a calculation. If for encryption equation E and calculation F have the correct calculation and make G become

$$E(K, F(x_1, ..., x_n)) = G(K, F, (E(x_1), ..., E(x_n))) \tag{1}$$

then the formula will consider that Encryption equation E for calculation F is Homomorphic [2].

However, if the equation only work for

$$\boldsymbol{F(x_1, ..., x_n) = \sum_{i=1}^{n} x_i} \tag{2}$$

Then it is considered as an Additively Homomorphic Encryption [2].

If the equation only work for

$$\boldsymbol{F(x_1, ..., x_n) = \prod_{i=1}^{n} x_i} \tag{3}$$

Then it is considered as a Multiplicatively Homomorphic Encryption [2].

Both Additively and Multiplicatively Homomorphic Encryption (including FHE) is very efficient for Ciphertext calculation [2].

2.2. FHE (Fully Homomorphic Encryption)

Fully Homomorphic Encryption (FHE) is also a type of encryption. Different with Partially Homomorphic Encryption (PHE), FHE can do much more complex calculations and have more variety of types to calculate it. For example, PHE can only do calculations with Addition and multiplication, but the FHE can do the same Addition and multiplication but also other mathematical logical calculations. FHE enables the performance of unrestricted calculations on encrypted data. For example, if a user has a function f and wishes to obtain the result of $f(m_1,\dots,m_n)$ for given inputs $m_1 \dots m_n$, they can execute the computation on the encrypted inputs $c_1 \dots c_n$. The final outcome remains in an encrypted state, but the user can employ their private key to decrypt the result and acquire the originally computed result, which is$f(m_1,\dots,m_n)$. This facilitates a wide range of computations on encrypted data without the necessity to decrypt it beforehand, thereby upholding data privacy and security [3].

2.3. Comparison

Even though both methods allow users to calculate the data without decryption, there is still some advantage and disadvantage for both methods. In terms of complexity, the calculation of FHE is usually more complicated and requires more time to process it. But PHE is easier and will take less time because it only calculates with two algorithms. It means that PHE has limited functionality and cannot perform many types of calculations. However, FHE can because it allows much more algorithms. Last point will be that complex calculations need more energy to keep it running, so obviously FHE's Computing overhead is much higher than PHE. To conclude this section, the choice of which method to use depends on the specific scenario and requirements. If users need to perform multiple complex computational calculations and keep your data private, FHE might be a better choice. If you only need to perform simple computational operations, PHE may be more suitable because of its higher performance.

3. Secure Multiparty Compute

In modern society, personal privacy data is constantly collected, at the most of the time personal data is shared by multiple parties. When one party has to share data with the other party, personal data is easily leaked, and Secure Multiparty Compute has its special algorithm to satisfy the needs of both sides. The result of this algorithm is one party no longer has to worry about information leakage, and the other party has access to the data to do things they want.

Secure Multiparty Compute allows different parties to use the same group of data to do calculation. But it can keep the data private at the same time. Secure Multiparty Compute provides privacy protection for collaborative computing. During the situation, security requires every participant only get information they need to make the output, they can't get further data. Actually, if the situation has several any computing task with multiple participants can use Secure Multiparty Compute. The paper [4] Protocols for secure multiparty computation (MPC) allow a group of participants to collaborate and compute a collective function based on their individual private inputs, ensuring that only the final output is revealed. An accompanying illustration illustrates the fundamental concept of this theory (Figure 2).

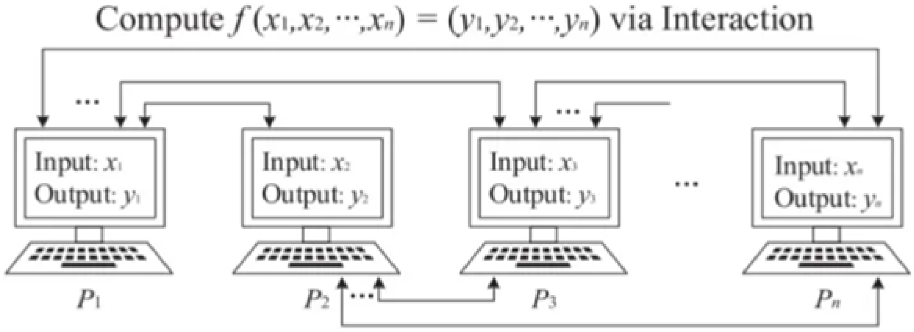
Compute $f(x_1,x_2,\cdots,x_n) = (y_1,y_2,\cdots,y_n)$ via Interaction
Input: x_1 Output: y_1
Input: x_2 Output: y_2
Input: x_3 Output: y_3
Input: x_n Output: y_n
P_1
P_2
P_3
P_n

Figure 2. Diagram of Secure Multi-party Computation [4]

SMC needs to consider the possibility of dishonest acts by dishonest people, the purpose of hacking may want to stole other people's private information. Someone attacks to crash the system. To solve this problem, threats and security requirements and security model are considered to prevent this behavior.

3.1. Threats and security requirements

Overall, secure multi-party computing provides a framework for secure and privacy-protecting collaborative computing between multiple parties, enabling them to work together while protecting confidential information. Scientists have come up with many definitions of security. There are five aspects mentioned in papers [5, 6].

Privacy: None of parties can get extra information expect the information they get at first and the result they get from the information. In other words, one party can only get their own input and output.

Correctness: The output received by each participant should be correct.

Input independence: The input chose by dishonest participant must be independent of the input of the honest participant.

Output assurance: Participants should not prevent other participants obtaining their own output information.

Fairness: When honest participants get their output, dishonest participants can start to receive their output.

These five are some requirements that a security protocol needs to satisfy. An ideal or reality simulation paradigm is given in paper [5]. In the context of SMC the ideal or realistic simulations paradigm refers to a framework that allows for secure computations among many parties and keep privacy and confidentiality at the same time. It involves simulating and testing computations in a controlled environment, ensuring the integrity of the data, and preventing unauthorized access. This paradigm aims to strike a balance between achieving realistic computational outcomes and maintaining strong security measures (Figure 3).

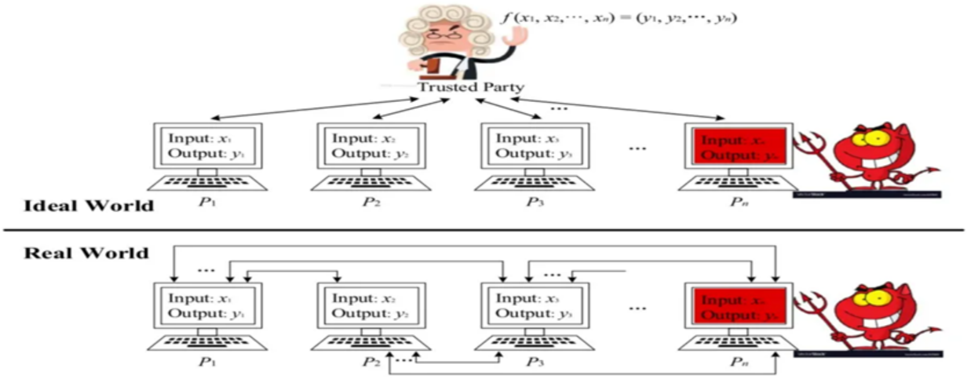
$f(x_1, x_2, \cdots, x_n) = (y_1, y_2, \cdots, y_n)$
Trusted Party
Input: x_1 Output: y_1
Input: x_2 Output: y_2
Input: x_3 Output: y_3
Input: x_n Output: y_n
P_1
P_2
P_3
P_n
Ideal World
Real World

Figure 3. Ideal/Real simulation paradigm [5]

3.2. Inadvertent transmission and its extension:

As we know from the previous content, the principle of SMC is that a participant sender sends data, and another receiver chooses among these data, after completing the transmission, the sender does not know what data the receiver has received, and the receiver has received the data, but he does not know other data besides the data he has selected. The aim of inadvertent transmission extension is to reduce excessive operating costs. Paper [6] was explained that the Inadvertent Transport extension protocol functions by concurrently operating multiple "base" Inadvertent Transport instances. The quantity of these "base" instances is determined by the chosen security parameters.

4. Encryption Technology

Data encryption technology, as a key means of information security, has the core idea of converting plaintext into ciphertext through specific algorithms, thus preventing unauthorised people from reading or modifying the ciphertext to ensure the confidentiality, integrity and accessibility of information. Data encryption can effectively protect all kinds of data files and make the data have high security and confidentiality.

4.1. The basic principles and types of data encryption technology division

(1) Symmetric key encryption

The main principle of symmetric key is to use different types of cryptographic algorithms to generate the same size, unequal and independent ciphertext information after data processing. This technology is a widely used encryption method in the field of computer network security. In the field of computer information security, symmetric key encryption algorithm is mainly used to scramble digital images to achieve the confidential transmission of digital image information. It is a technique that uses the same key for encryption and decryption, and it is considered to be one of the most widely used encryption methods at present. Compared with asymmetric key encryption methods, symmetric key encryption techniques have higher encryption speeds and are particularly suitable for occasions where the amount of data is relatively small [7].

The Figure 4 shows the principle of the symmetric key process

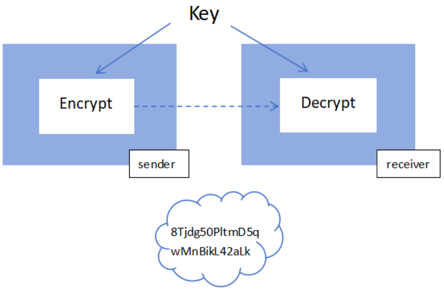

Figure 4. The symmetric key process [7]

Common symmetric key encryption techniques include Data Encryption Standard, TripleDES, Advanced Encryption Standard, etc. DES is considered to be one of the earliest emerging symmetric key encryption techniques, which possesses excellent security features, but is susceptible to brute-force cracking attacks because the length of its key is only 56 bits. Hence, the 3DES algorithm was introduced in response. This encryption algorithm uses three 56-bit keys to encrypt data, thus enhancing data security.AES algorithm is considered one of the most commonly used symmetric-key encryption techniques, and its key lengths can reach 128, 192, and 256 bits, which significantly improves its security, and it is very suitable for a variety of application scenarios.

(2) asymmetric key cryptography

Asymmetric key encryption technology is mainly used in e-commerce, e-government and other fields. This technology, as a key encryption means, has a more complex encryption process compared to symmetric key encryption technology due to the use of different types of keys in the encryption and decryption process, but at the same time, it also improves the overall level of security. Asymmetric key encryption is widely used in various information security fields because it requires less security for the key. Asymmetric key encryption methods require the use of two keys: a public key and a private key. The public key is publicly available, but the private key is restricted to the holder of the key. Therefore, this type of encryption is often used in scenarios where keys need to be distributed.

RSA is the most widely used asymmetric encryption algorithm.The principle of the RSA public key cryptosystem is based on number theory, which takes advantage of the fact that it is easy to multiply two large prime numbers and extremely difficult to factorize their product. Therefore, the product can be used as a public encryption key [8].

$$\text{Ciphertext} = \text{Plaintext}^{E} \bmod N$$

$$\text{Plaintext} = \text{Ciphertext}^{D} \bmod N$$

The encryption and decryption in RSA use the same number N. The fact that the public key is public means that N is also public. So the private key can also be thought of as just D.

(3) Hash calculation

The algorithm is also used in other applications such as e-commerce transaction security as well as network information security. The technique is a key means of data encryption and its core function is to compress a message of arbitrary length into a fixed-length digest to ensure data confidentiality and integrity. Hash algorithms have gained wide application and acceptance in the field of cryptography and network security due to their unique attributes of irreversibility, uniqueness and tamperability. The hash function is one of the most efficient ways to implement encryption and decryption functions, and the algorithm has become one of the most widely used encryption and decryption algorithms today due to the less time required to use the hash function. Irreversibility means that the digest cannot restore the original message, so the digest does not reveal any information about the original message, while uniqueness means that different messages generate different digests, which ensures the uniqueness of the digest. The hash algorithm also possesses the property of immutability, which means that the generated digest is immutable, and once the data is changed, the digest will be modified accordingly to ensure the integrity of the data (Figure 5) [9,10].

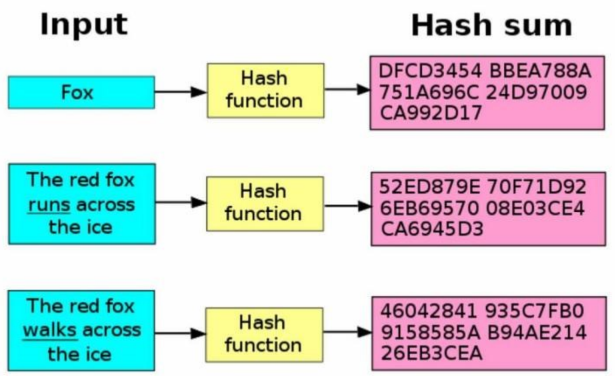
Input
Hash sum
Fox
Hash function
DFCD3454 BBEA788A 751A696C 24D97009 CA992D17
The red fox runs across the ice
Hash function
52ED879E 70F71D92 6EB69570 08E03CE4 CA6945D3
The red fox walks across the ice
Hash function
46042841 935C7FB0 9158585A B94AE214 26EB3CEA

Figure 5. Hash calculation [9]

5. Conclusion

This article summarized the methods for preventing user privacy leakage, with a focus on discussing the principles of homomorphic encryption and the method of accessing data without the need for decryption through Homomorphic Encryption (HE). HE is an encryption technology that allows calculations on ciphertext without revealing the plaintext data. Additionally, the article introduced the principles of secure multiparty computation, which enables multiple users to share and compute data while preserving privacy. Furthermore, data encryption technology, using specific algorithms to convert plaintext data into ciphertext, was addressed to ensure data consistency and privacy.

To further explore, as data and information continue to grow rapidly, the risk of user privacy leakage becomes more severe, making privacy protection an ongoing and important challenge.

In terms of homomorphic encryption, although progress has been made, there is still room for improvement in terms of performance and practicality. Future research and development will focus on improving the efficiency of homomorphic encryption algorithms and exploring broader application areas to better meet practical needs.

The concept and technology of secure multiparty computation will continue to evolve and be more widely adopted. Currently, secure multiparty computation is often deployed in trusted environments, but it poses challenges in open network environments. Future research may focus on designing secure multiparty computation protocols that are adaptable to open network settings, ensuring privacy protection for users in untrusted environments.

Additionally, data encryption technology will continue to evolve to meet the changing security threats and demands. New encryption algorithms and protocols will be developed to counter more complex and advanced attack techniques while maintaining data consistency and integrity during the encryption process. In summary, with the increasing demand for privacy protection, we can expect continuous development and innovation in homomorphic encryption, secure multiparty computation, and data encryption technologies. Strengthening user education and awareness will also play a key role in protecting user privacy and promoting the acceptance and adoption of privacy protection technologies.

Authors Contribution

All the authors contributed equally and their names were listed in alphabetical order.

References

[1] Acar, A., Aksu, H., Uluagac, A. S., & Conti, M. (2018). A survey on homomorphic encryption schemes: Theory and implementation. ACM Computing Surveys (Csur), 51(4), 1-35.

[2] Li Shun-Dong, Dou Jia-Wei, & Wang Dao-Shun. (2015). Homomorphic encryption algorithm and its application in cloud security. Journal of Computer Research and Development, 52(6), 1378-1388.

[3] Armknecht, F., Boyd, C., Carr, C., Gjøsteen, K., Jäschke, A., Reuter, C. A., & Strand, M. (2015). A guide to fully homomorphic encryption. Cryptology ePrint Archive.

[4] Lindell, Y. (2020). Secure Multiparty Computation (MPC). EPrint IACR.

[5] Zhao, C., Zhao, S., Zhao, M., Chen, Z., Gao, C.-Z., Li, H., & Tan, Y. (2019). Secure Multi-Party Computation: Theory, practice and applications. Information Sciences, 476, 357–372.

[6] Knott, B., Venkataraman, S., Hannun, A., Sengupta, S., Ibrahim, M., & van der Maaten, L. (2021). CrypTen: Secure Multi-Party Computation Meets Machine Learning. Neural Information Processing Systems; Curran Associates, Inc.

[7] Wang Fan,Qinrang Liu,Xinyi Zhang,Yanzhao Gao,Xiaofeng Qi & Xuan Wang.(2023).A Symmetric and Multilayer Reconfigurable Architecture for Hash Algorithm. Electronics(13).

[8] Liu Guanxiu.(2022).The Application of Data Encryption Technology in Computer Network Communication Security. Mobile Information Systems.

[9] Chen Lin.(2021).Application of Computer Network Communication Technology in Production and Life. Journal of Physics: Conference Series(3),032161-.

[10] Hu Xiaoling.(2020). Application of the Data Encryption Technology in the Computer Network Communication Security. Basic & linical Pharmacology & Toxicology,283-283.