Sri Lanka Institute of Information Technology

Homomorphic Encryption

Individual Assignment

IE2022 - Introduction to Cyber Security

Submitted by:

Student Registration Number	Student Name
IT1902914	J C Hapuarachchi

Date of submission:26/4/2020

Table of Contents

Abstract

This report was based on Homomorphic Encryption. An online research was done to gather the information required for the report. Give a basic idea about Homomorphic Encryption, the evolution of Homomorphic Encryption and future developments of Homomorphic Encryption were the main objectives of this report. Apart from the main objectives, to find out how Homomorphic Encryption works, where Homomorphic Encryption can be applied and limitations of Homomorphic Encryption can be identified as the sub-objectives of this report. In this research a qualitative design was used and online websites, blogs and videos were used to gather the information which was required to the report. The required information was gathered from 7 different websites and blogs. Apart from these websites and blogs, a video in YouTube was used to gather the required information. The report was mainly divided in to three parts. "Introduction of Homomorphic Encryption", "Evolution of Homomorphic Encryption" and "Future developments of Homomorphic Encryption" were the main three parts of this report. This report was mainly developed to give a basic idea about Homomorphic Encryption. First the meaning of Homomorphic Encryption was given and then the how Homomorphic Encryption works, application of Homomorphic Encryption and the disadvantages of Homomorphic Encryption was revealed respectively. After the introduction, Evolution of Homomorphic Encryption was revealed. The evolution of Homomorphic Encryption was revealed from the beginning of Homomorphic Encryption up to present day Homomorphic Encryption technologies. All these evolutionary stages were clearly explained in the report. The importance of Homomorphic Encryption was revealed to the outside world through this report. The advantages customers can benefit and how customer's privacy can be protected were also revealed through the report. Not only the advantages but also the disadvantages were also revealed from this particular report. After the introduction, then history of Homomorphic Encryption was revealed. From the history it was revealed that though the idea of Homomorphic Encryption was proposed long time back, still this encryption technology is on the developing stage. The things done by well-known companies such are Microsoft and IBM were revealed from this report. The expected future improvements in the field of Homomorphic Encryption was revealed in the last part of the

report and the importance of Homomorphic Encryption was revealed throughout the entire report.

1. Introduction to Homomorphic Encryption

Before going to Fully Homormophic Encryption, first we need to know what Encryption is.

What is Encryption?

We can simply define Encryption as a process. It is a process which converts data in plaintext which is a readable format to a cipher text which is an unreadable format. After encrypting, plain text becomes cipher text. To convert a plain text to a cipher text a particular formula is used. These formulas are known as algorithms or ciphers. Ciphers is a combination of encryption and decryption algorithms. In the cipher there is a variable. This variable is used to store the most important thing in the encryption process, which is the key. The encrypted data can only be decrypted using this key. Only the people who knows the key will be able to decrypt the data.

Now let's look at how encryption works. In the beginning of the encryption process, the user must decide what cipher he/she will use to encrypt their data.
There are two widely used ciphers. They are,

- Symmetric cipher
- Asymmetric cipher

Symmetric cipher uses only a single key in its encryption process. The same key is used to encrypt the data as well as decrypt the data. This key is sometimes known as shared key. This is referred like this because the sender who encrypts the data should share the key with the other people who are authorized to decrypt the encrypted data. On the other hand Asymmetric cipher uses two different keys to encrypt and decrypt. The two keys which are used is private key and public key. Any of these keys can be used in encrypting data or decrypting data. However, the key which is used to encrypt data will not be used to decrypt data. The keys which are used in encryption process are generated using random number generator or computer algorithms.

Encryption can be simple define as a process which is used to securely transit data between entities. Encryption helps in preserving Confidentiality, Integrity and Availability [CIA] of data. Without encryption CIA would be violated. Encryption protects data from

unauthorized users and gives only access to the authorized users for the encrypted data. Encryption is commonly used to protect data at transit as well as protect data at rest. Encryption is one of the main important concepts when it comes to world of Cyber Security.

Now let's move on to Homomorphic Encryption.

What is Homomorphic Encryption?

Homomorphic Encryption allows parties to perform computations on encrypted data without decrypting it. This is a new type of encryption. Using this, computations can be done on encrypted data with decrypting the data and also the parties who do the computation need not to know the key which is needed to decrypt this. The results of this computations will be in encrypted data which is in an unreadable format and this encrypted results can only be read by the authorized people who know the key.

Homomorphic encryption is similar like all other encryption mechanisms. The main purpose of this encryption mechanism is same like all other encryption mechanisms which is to preserve Confidentiality, Integrity and Availability of data, but there is a difference in this encryption mechanism when comparing with other encryption mechanisms. The similarity of this encryption mechanisms with other encryption mechanisms is this uses a public key to encrypt data and only the authorized people who have the key will be allowed to access the encrypted data. The difference is that this mechanism uses an algebraic system which allows the parties to perform computations on the encrypted data. In real world practice, this encryption mechanism works best with data which is represented using integers. The main two computational functions which can be done on this encrypted data are addition and multiplication. In other words this means that computations can be done on encrypted data just like when it is in plain text. The computations of Homomorphic Encryption are represented as either Boolean or arithmetic circuits. The parties can perform computations on these encrypted and the results will also be in encrypted format and the results can only be decrypted by the people who have the key.

Homomorphc Encryption uses different types of schemes to perform computations on encrypted data. Following are the some examples of the schemes which were used in Homomorphic encryption,

- RSA Algorithm
- ECC encryption
- ELGamal cryptosystem
- Pailler cryptosystem

Homomorphic Encryption is divided into 3 Homomorphic Encryption types. They are,

- Partially Homomorphic Encryption
- Somewhat Homomorphic Encryption
- Fully Homomorphic Encryption.

The difference between the above types of Homomorphic Encryption are the different types and frequencies each type of Homomorphic Encryption have when performing computations or operations on encrypted data.

- Partially Homomorphic Encryption

In this type of Homomorphic Encryption, only one selected computation or operation can be done on encrypted data. This operation can either be addition or can be multiplication. This selected computation can be done on the encrypted data unlimited number of times.

- Somewhat Homomorphic Encryption

This is just like Partially Homomorphic Encryption, but in this type the multiple operations can be performed on encrypted data. The drawback of this type is that the operations can be performed only a limited number of times.

- Fully Homomorphic Encryption

This was developed from Somewhat Homomorphic Encryption type. In this type,

Addition and multiplication computations can be done on encrypted data for any number of times as required. The other Homomorphic Encryption types cannot handle arbitrary computations on cipher text, but in this type it can be handled.

The need of Homomorphic Encryption

Before Homomorphic Encryption was found, when there is a need to perform computations on encrypted data, first the encrypted data had to get decrypted. To perform computations on the encrypted data, end users get the help from the cloud service providers and as well as from some other third parties. These cloud service providers and these third parties perform computations at a lower cost and end users find it easy to get the help from third parties rather than handling data themselves. For the third parties to perform computations on the encrypted data first data needed to get decrypted. When encrypted data gets decrypted by third parties, these data become vulnerable to unauthorized people and also when data gets decrypted by the third parties, this will affect the privacy of the end users. Furthermore these data can get exposed to data breaches. If the sensitive data goes to the hand of bad person, the person can do various bad things using this data. So the data should not be decrypted even by the third parties or cloud service providers that helps in performing computations on the data.

Simply we can define Homomorphic Encryption as a technique which can be used to preserve the privacy of end user when the third parties are handling computation on end user's data.

How is Homomorphic Encryption done?

The Homomorphic Encryption have five stages. The five stages are,

1. Set up the scheme, security parameters and functionality parameters.
2. Key generation – The keys which are used in this encryption are,
    - Secret key
    - Public key
    - Relinearization key
    - Galois key
3. Encryption – a number or a set of numbers are transformed into cipher text which is in two polynomials.
4. Evaluation
5. Decryption

The first step is the set up step. In this step you need to choose the scheme you are going to use, the security parameters and the functional parameters or performance parameters.

At present day, Homomorphic Encryption there are mainly three types of schemes. They are,

- TFHE : logic gates on bits
- BGV, BFV: perform exact arithmetic on vectors of numbers
- CKKS: perform approximate arithmetic on vectors of numbers

The next stage is key generation. As we all know keys are very important when it comes to encryption and in this step, the keys are generated. Secret key is a very important key because it is the key which is used to decrypt the particular encrypted data. The secret key need to be protected and it should be stored securely without getting to the hands of attackers or bad people. Public key, Relinearization key and Galois key is generated from the secret key and all these three keys are considered as types public keys. Relinearization and Galois key get used in computations and Public key is used in encryption. The third stage is the encryption stage. In this stage data which is in plain text (a polynomial) is converted into cipher text (2 polynomial).The next stage is the evaluation stage. This is the

stage where the program is written with the required computations. This stage is the most important stage in Homomorphic Encryption. Actually this is the stage where Homomorphic is actually performed and the last stage is the decryption of the results. This decryption is done only by the people (the parties who knows the key) who are authorized to decrypt the data.

Practical application of Homomorphic Encryption

Encrypted databases

Database is considered as the soul of any organization or business. Database is simply a collection of data. This data can be confidential data, financial data and all other data types. Database is considered as one of the most important components in the IT infrastructure. This database need to be protected from the unauthorized people. Security is considered as one of the major concerns when it comes to database. To secure the database, various security techniques are used. One of the security techniques that is used is encryption. By using this, the database will be encrypted and data will not be understood by the bad people even in an event like data breach. Though from this technique data is protected, search queries or computations cannot be performed on encrypted databases. Simply it means computations cannot be done on encrypted data. To search queries in the database, the data need to be in decrypted format. When the data is in decrypted format, the database gets vulnerable to attacks. The solution for this is Homomorphic Encryption. Using Homomorphic Encryption computations and search queries can be done on encrypted data.

Outsourcing computation on the cloud

Cloud computing have become very popular during the recent years. The popularity of cloud computing rose because it provides all the necessary infrastructure to perform operations, computations and so on. The users tend to use cloud computing more because the infrastructure which end users expect will be provided to them at a low cost. End users uses cloud service providers to perform computations on their data. When performing computations on the encrypted data, the data need to get decrypted, because of this the end user should trust the cloud service provider. The data of a particular user is something which is private to them and when encrypted data get disclosed to cloud service providers, it will cause some problems regarding the privacy of the end user. As a solution for this Homomorphic Encryption is used. The use of Homomorphic Encryption enables the cloud service providers to perform computations on encrypted data without decrypting it and the results will only be visible to the end user.

Improving Election Security and Transparency

When counting the votes, the count is visible to the parties who perform the count. These parties who did the computation becomes aware about the count. This too can be prevented using Homomorphic Encryption. If Homomorphic encryption is used, the results will be in an unreadable format and only the authorized people who have the key will be able to see the results.

Protection of mobile agents

Homomorphic Encryption is very useful in the world of mobile phones. The mobile phone contains functions and data. These functions and data are confidential for that particular mobile owner. This confidential data should not be disclosed other people. Otherwise it will cause problems regarding the privacy of the mobile owner. As a solution for this Homomorphic Encryption can be used. Using Homomorphic Encryption computations can be done on encrypted functions and encrypted data. If this method is used it will not cause problems to the privacy of the mobile user because mobile data won't get disclosed to other parties.

Protecting the privacy of patients in medical sector

As an example when it comes to medical sector, predictive analytics are done using patient information. This information contains confidential information of the patient. But when doing predictive analytics this data gets disclosed to predictive analytic service providers, when this data get discloses to these parties it will affect the privacy of the particular patient. As a solution for this Homomorphic Encryption can be used. Using this encryption type the predictive analytic service providers can do the predictive analytic using the patient's data when the patient's data is in encrypted format and this confidential data won't get disclosed to these service providers because this data. Not only this, Homomorphic Encryption can be used to do many things in the medical sector without affecting the patient's privacy.

Other Applications

In the recent years, machine learning and artificial intelligence have become very popular. Different types of industries and enterprises uses machine learning and artificial

intelligence to improve the quality of the service they provide. Machine learning and artificial intelligence build the statistical models with the available data. The industries contains employee data and also customer data. When these data was used to build statistical models it will be problems to the privacy of the customers as well as the privacy of the employees. As a solution for this Homomorphic Encryption can be used.

Drawbacks and limitations of Homomorphic Encryption

- Computational speed is slow

One of the major drawbacks of Homomorphic Encryption is its computation speed. The speed of doing computations on the encrypted data is much slower than when computations are performed on decrypted data. Approximately when the computations are performed on encrypted data, it is 106 times slower than performing computations on decrypted data. The reason why Homomorphic encryption is slow, is because of the bootstrapping step. This applies an additional operation to cipher ext. This additional operation is known as Recrypt. By recrypting the complexity of the system will be increased. Because of this additional operation the speed of performing computation in Homomorphic Encryption is slower than performing computations on decrypted data.

- Limited number of operations

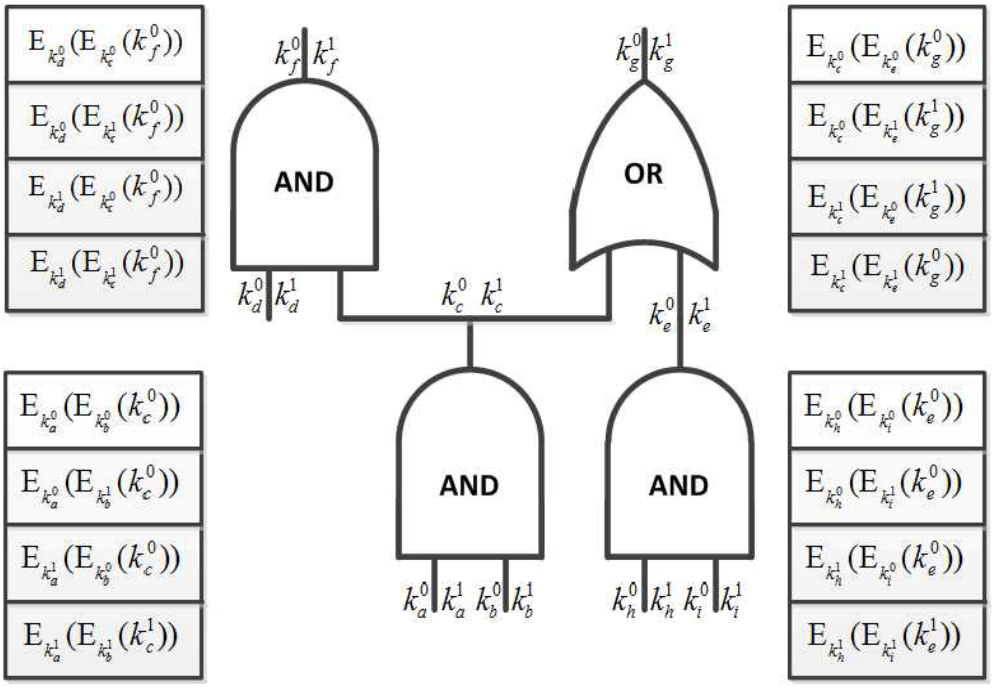
Another drawback is the number of limited operations that can be done on encrypted data. As an example, only the computing statistical functions such as finding the mean, variance and standard deviations are feasible in Homomorphic Encryption.

2. Evolution of the topic

The word “Homomorphic” in Homomorphic Encryption came from the word “Homomorphism”. The word “Homomorphism” came from the word “Homomorphe”. In Greek “Homo” means “same” and “morphe” means “shape”. In abstract algebra Homomorphism is defined as a map which contains all the algebraic structures between the domain and the range of an algebraic set. This map is an operation. It is an operation which takes inputs from the domain and gives an output which is an element in the range. This operation can be addition, multiplication and etc. But when it comes to world of Cyber Security Homomorphic is an encryption type. It is an encryption type which is still in the developing stage. Though it is still developing the idea of Homomorphic Encryption was proposed few years back.

The idea of Homomorphic encryption was first proposed in 1978. This idea was proposed just a after a year of introducing RSA public key scheme. This idea was first proposed by Ron Rivest, Len Adleman, and Michael Dertouzos. They proposed this idea through a report namely, “On Data Banks and Privacy Homomorphisms”. In this report they used an example of a loan company to propose their idea. They said that a particular loan company can use a cloud service provider to perform computations on encrypted data. This idea was lead to the term called “Homomorphic Encryption”.

In 1982 a person named Yao made an attempt to try Homomorphic Encryption. This is known as Yao’s garbled circuit study. Using this Yao tried to perform computations on encrypted data. To propose his idea he used a problem of rich people where Homomorphic encryption can be used. The problem was to compare the wealth of two rich people without revealing the amount of money to each other. As a solution for this problem he proposed the two party communication protocol. But there was a drawback in Yao’s garbled circuit solution. The problem was that the size of the cipher text increases in each computation. Because of this the efficiency in computation was not that much in a good standard.



A simple example of Yao's garbled circuit

Homomorphic Encryption is developed using different kind of schemes which is found during different years. When considering about Homomorphic Encryption there are four types of generations. They are,

1. Pre Fully Homomorphic Encryption
2. First generation Fully Homomorphic Encryption
3. Second generation Fully Homomorphic Encryption
4. Third generation Fully Homomorphic Encryption
5. Fourth generation Fully Homomorphic Encryption

Pre Fully Homomorphic Encryption is considered as the start of Homomorphic Encryption. After Pre Fully Homomorphic Encryption, First generation Fully Homomorphic Encryption came into being. Gentry's schemes was used in this to do the computations. First generation Fully Homomorphic Encryption was first started with Somewhat Homomorphic Encryption. As we already know Somewhat Homomorphic

Encryption can perform only a number of operations. As a solution for this Gentry showed that any Somewhat Homomorphic Encryption can be turned into Fully Homomorphic Encryption. He said that Somewhat Homomorphic encryption can be turned into Fully Homomorphic Encryption through recursive self-embedding. Gentry's scheme's the computations are done using ideal lattices. According to Gentry Fully Homomorphic Encryption have few steps. The steps are,

1. First a somewhat Homomorphic Encryption scheme is constructed, this is designed to support low degree polynomials on encrypted data.
2. Next the decryption procedure is squashed, then it is expressed as a low degree polynomial which is supported by the scheme.
3. Then at the last step, a bootstrapping transformation is applied in order to get a fully homomorphic scheme.

In 2010 a second Fully Homomorphic encryption scheme was presented. It was presented by Marten Van Dijk, Craig Gentry and Shai Halevi. These three showed that the somewhat Homomorphic Component which is in Gentry's ideal lattice based scheme can be replaced with a very simple somewhat Homomorphic scheme which uses simply integers. Because of the use of this simple somewhat Homomorphic scheme, it was much simpler than the Genry's ideal lattice based scheme. Though this scheme is simpler, by property wise it is similar to the Genry's ideal lattice based scheme and can perform similar operations. In 2010 another thing happened. It is that in 2010 researchers examined way of how to improve the key generation scheme in Homomorphic Encryption. One of the researchers namely Ogura et al proposed of generating keys from random elements and analyzing eigenvalues of the corresponding keys. However, this was not that much successful, because this proposed method by Ogura et al was unable to achieve the efficiency which was expected.

In 2012 a person named Chunseng proposed a modification for Fully Homomorphic Scheme. Chunseng applied a self bootstrappable technique to the Fully Homomorphic scheme, and because of this the security was only depended on the hardness of the polynomial cosset problem.

Second generation Fully Homomorphic Encryption was developed from the schemes which was began to develop starting from 2011-2012 by Zvika Brakerski, Craig Gentry, Vindo Vaikuntanathan and some others.

The schemes which are uses in Second generation Fully Homomorphic Encryption are,

- The Brakerski-Gentry-Vaikuntanathan (BGV, 2011) scheme, building on techniques of Brakerski-Vaikuntanathan.
- The NTRU-based scheme by Lopez-Alt, Tromer, and Vaikuntanathan (LTV, 2012)
- The Brakerski/Fan-Vercauteren (BFV, 2012) scheme, building on Brakerski's *scale invariant* cryptosystem
- The NTRU-based scheme by Bos, Lauter, Loftus, and Naehrig (BLLN, 2013), building on LTV and Brakerski's scale-invariant cryptosystem;
- The Cheon-Kim-Kim-Song (CKKS, 2016) scheme.

The Second generation Fully Homomorphic Encryption also followed the basis of Gentry's original construction. Which means that first they develop a Somewhat Homomorphic cryptosystem and then it is converted into Fully Homomorphic cryptosystem.

The third generation Fully Homomorphic Encryption was developed based on the schemes which was develop in 2013 by Craig Gentry, Amit Sahai and Brent Waters. When doing Homomorphic encryption there is a step which costs more money. This step is "relinearization". Due to the schemes which was used in third generation Fully Homomorphic Encryption this expensive step was avoided and not wanted.

The bottlenecks of Homomorphic Encryption are in bootstrapping and large cipher expansion. In 2015, Ducas and Micciancio the computational operations can be in done in the speed of less than a second and then in 2016 Chilotti et al reduced this computational operation speed up to 13ms.

In 2015 IBM released its first version of Homomorphic Encryption libraries in C++. But unfortunately the version which was released by IBM was not up to the required standard in terms of performance. But in 2018 IBM released another version Homomorphic Encryption and they called this as," re – implementation of homomorphic linear transformation". This version was between 15 and 75 times fast. In 2019 IBM successfully carried out an experiment with a large bank to perform homomorphic encryption or in other words IBM managed to perform computations on bank's encrypted data. In 2019 Microsoft also released Microsoft SEAL. Microsoft SEAL was powered by open source homomorphic encryption technology. SEAL provides a set of libraries which allows to do computations or operations on encrypted data.

3. Future developments in Homomorphic Encryption

In the next few years we all hope that there will be a good development in Homomorphic Encryption. In Homomorphic Encryption there are a lot of things to get developed.

When talking about cloud service providers, though Homomorphic Encryption is an important technology to them, all the cloud service providers do not use it. Only some of the cloud service providers uses this technology. This might be because of the high cost when implementing Homomorphic Encryption. But in the near future there is a high chance that almost every cloud service providers will get adopted to use this Homomorphic Encryption technology. The cloud service providers need to implement the right security standards and it is obvious that Homomorphic Encryption will be able to increase their security standard. Though Fully Homomorphic Encryption is not always applied to real world scenarios, cloud service providers, still can use Partially Homomorphic Encryption and it will benefit the customers who use their services and the trust the customers have towards cloud service providers will get increased.

When it comes to the perspective of cloud service providers there is another problem. The problem is not for the customers who uses the services of cloud service providers, but for the cloud service providers. Most of the time cloud service providers earn money by selling the customer information to other parties. But if customer data is in encrypted format, then the cloud service providers will not be able to sell it to the third parties and earn money, then this will disrupt one of the earning money methods of the cloud service providers. When cloud service providers couldn't get money from the other parties, then they will request a pay from the customers for the service they provide. But the problem is that though the customer thinks about the security of their data, will the customers pay money for the cloud service providers. Usually most of the customers prefer to get the service from the cloud service providers for free of charge. This is another current problem that have with Homomorphic Encryption and in the future it is expected that this problem will be also solved.

When it comes to Fully Homomorphic Encryption, there are still some problems with it and there are places where advancements are required. As an example though Homomorphic Encryption support multiple operations, still there are limitations when doing this. In the future it is expected that, these limitations will be solved and a more developed Homomorphic Encryption technology will be handed over to the customers in the near future.

Another problem with Homomorphic Encryption is that it is expensive to implement and it is not practically inefficient. At present researches have been done to improve the efficiency of Homomorphic Encryption and in the next few years Homomorphic Encryption will become more efficient and there is a chance the cost of using Homomorphic Encryption will be reduced.

Since the most well-known companies like IBM and Microsoft have already focused their attention on Homomorphic Encryption, there is a high chance that these companies will help in improving this technology greatly.

4. Conclusion

This report was done based on Homomorphic Encryption.

First, the meaning of Homomorphic Encryption was showed to the people who read this report. This was included in the part," Introduction to Homomorphic Encryption". Homomorphic Encryption was revealed as a very important technology from the report. It was revealed that Homomorphic Encryption can be used to protect the customer's privacy and also it was revealed that if third parties such as cloud service providers use Homomorphic Encryption, the customer's trust would be increased. Apart from the advantages, from the disadvantages specified in the report, it was revealed that the slowness of the computation speed when performing computations on encrypted data was one of the main drawbacks of Homomorphic Encryption. Not only this, the fact that only a limited number of operations can be performed on encrypted data was also another major drawback.

From the history it was revealed that though the idea of Homomorphic Encryption was first proposed in 1978 by Ron Rivest, Len Adleman, and Michael Dertouzos, still it is in the developing stage. Though the companies like IBM and Microsoft had focused their attention on Homomorphic Encryption, this technology was still not up to the expected standard. From this report it was revealed that many companies like Microsoft and IBM are trying their best to get Homomorphic Encryption up to the required standard. Apart from that not only the companies but also the individuals were also trying their best to do experiments on Homomorphic Encryption.

From this report it was revealed that, most of the people thinks that the drawbacks which Homomorphic Encryption have today, will be solved in the near future. The expectations that people have towards Homomorphic Encryption which were revealed from the report were the cost of using Homomorphic Encryption would be decreased and also the efficiency of Homormorphic Encryption would be increased. If all these drawbacks are minimized to a lower level, then the Homomorphic Encryption would become one of the most widely used encryption technology. From this report it is revealed that Homomorphic Encryption was considered as one of the most important technologies to cloud service providers. Though it is important, not all cloud service providers use this

technology. As said by the report we all expect that all these problems will be solved in the near future

5. References

Homomorphic Encryption – Theory and Application: Intechopen , April 16.[Online]. Available:https://www.intechopen.com/books/theory-and-practice-of-cryptography-and-network-security-protocols-and-technologies/homomorphic-encryption-theory-and-application

What is Encryption & how does it work: Medium, April 16. [Online]. Available:https://medium.com/searchencrypt/what-is-encryption-how-does-it-work-e8f20e340537

Microsoft Research. *Intro to Homomorphic Encryption* (Dec 19, 2019).Accessed: April 16 [Online]. Available: https://www.youtube.com/watch?v=SEBdYXxijSo&t=1262s

What is Homomorphic Encryption? And why is it so transformative? : Forbes, April 16 [Online]. Available:https://www.forbes.com/sites/bernardmarr/2019/11/15/what-is-homomorphic-encryption-and-why-is-it-so-transformative/#5a0c2b267e93

Provenance for cloud data accountability: Science Direct, April 16 [Online]. Available:https://www.sciencedirect.com/science/article/pii/B9780128015957000082

A guide to Homomorphic Encryption: Science Direct, April 16 [Online]. Available:https://www.sciencedirect.com/topics/computer-science/homomorphic-encryption

The advantages of disadvantages of Homomorphic Encryption: Baffle, April 16 Online]. Available:https://baffle.io/blog/the-advantages-and-disadvantages-of-homomorphic-encryption/

The fact and fiction of Homomorphic Encryption: Dark Reading, April 17 [Online].Available:https://www.darkreading.com/attacks-breaches/the-fact-and-fiction-of-homomorphic-encryption/a/d-id/1333691