Project Title: Homomorphic Encryption

Name of the Student: Carlos Gross-Martinez

Abstract

Homomorphic encryption is considered a new segment in the cryptography field which in its most basic notion, this cryptosystem's main purpose is to deliver the ability to conduct calculations in data that is encrypted without the need to decrypt it before performing the desired operation. Brakerski delivers a touching remark on the system by conveying that "This fully homomorphic encryption scheme which is quite beautiful by itself can actually be seen as just one phase of a really beautiful mathematical structure that can be used to achieve other exciting cryptographic goals" (Brakerski 00:35-00:48). This "mathematical structure" as defined by Brakerski was first made possible by Gantry with the introduction of his PhD thesis titled "A FULLY HOMOMORPHIC ENCRYPTION SCHEME". With this work, Gentry ends years of uncertainty regarding the feasible application of the idea. The concise overall mechanism of the cryptosystem consists of implementing a lattice based mathematical approach for creating the public and private key parameters which will be used to encrypt and decrypt the data to be transmitted for computations. Once the data has been encrypted and the public parameter have been made available, the encrypted cyphertext is sent to the server hosting the function that has the capability and design to perform the operation on the encrypted data. Once the server completes processing the data, the final result is considered to be a fully homomorphic encrypted cyphertext which can only be decrypted by the private key which was used in lattice based mathematical approach; also known as the lattice encryption scheme. This can be thought as the complete process having two levels or layers of encryption. On the first layer, the user encrypts the data before sending it out to the server, and in the second layer the server executing the operation on the data also encrypts the results after conducting the function or operation. The popularity in this cryptosystem is based on its key feature of privacy, since the true contents of the data are only known to the person who is encrypting it. This is also one of the main reasons why the improvement and advancement of this system is so important and sought after. Another great feature to mention from the fully homomorphic encryption scheme is that the encrypted cyphertext resultant from the execution of the computation by the server will not be bigger in memory size when compared with the size of the cyphertext encrypted by the user requesting the operation. There are multiple practical implementations of this encryption scheme which continue to feed in its growing popularity. In a general and practical implementation idea of the cryptosystem, Brakerski communicates that "if you do need the motivating example we can think about this problem of outsourcing computation where we have this weak device that wants to use a strong server in order to compute some function "(Brakerski 01:07-01:20). This practical implementation denoted by Brakerski is also inferred by Player when she states that this cryptosystem "is typically explained in the client-server model" (Player 00:54-00:55). It can be seen from these statements that the best scenario to implement the cryptosystem will necessitate to meet certain criteria to induce its implementation. By implementing this cryptosystem, it is not only possible to perform operations in unknown encrypted variables, this cryptosystem's robust architecture also provides a high level of security even when matched against the new emergent threats like quantum computers. This new and developing computing systems are capable of easily breaking some of the current encryption schemes which normally would take years for a regular computer to crack. In other words, the application of the fully homomorphic cryptosystem in an open communications channel guarantees high level of security and confidentiality of the data independent of the computer system that is trying to illegally acquire the information. This is a key beneficial piece or factor in the fully homomorphic cryptosystem since the dawn of the novel quantum computing era has endangered and made obsolete many of the current cryptographic protocols used for secure data transmission and communications. Based on this information, it also possible to understand why implementing the fully homomorphic encryption scheme through a public transmission channel will result in a more secure and discrete approach when compared with other protocols.

Introduction

Homomorphic encryption, also known as fully homomorphic encryption or FHE, is a novel segment in the cryptography field which in its most basic notion, this cryptosystem's main purpose is to deliver the ability to conduct calculations in data that is encrypted. Even though the application of this cryptographic approach is considered fairly new with its practical introduction with a little over a decade ago, the general idea of homomorphic cryptography has been around since 1978 when Adleman, Dertouzos, and Rivest first introduced its conception. Nevertheless, it wasn't until 2009 from its introduction that Craig Gentry presents the first practical structure of this conception. Gentry's homomorphic architecture is considered to be the first generation of FHE as a cryptosystem, and with this scheme, he eliminates the lingering uncertainty of a little over three decades, on whether or not, the solution to the idea of homomorphic encryption was possible. Since Gentry's scheme introduction, FHE has continue to evolve by developing new techniques and schemes to increase its performance and still deliver the same level of security and privacy. Moreover, there is an ongoing effort by different departments to create an industry standard for the emergent technology. In a more technical definition of the system, Wikipedia states that "Homomorphic encryption is a form of encryption with an additional evaluation capability for computing over encrypted data without access to the secret key. The result of such a computation remains encrypted". It is important to note that result of the encrypted computation can only be revealed to the owner of the secret key which encrypted the data. Moreover, homomorphic encryption comprises different models based on the category. The most common categories are fully homomorphic which allows for the evaluation of circuits with unbounded depth, leveled fully homomorphic which allows for the evaluation of circuits bounded by depth, partially homomorphic which allows for the evaluation of circuits comprising of one gate, and the somewhat homomorphic which allows for the evaluation of two gates with the constrain that only it is possible to conduct the valuations in a subsection of the circuit. The first implementation of FHE accomplished by Gentry consisted of a two-stage approach to the problem where on the first stage, he utilizes a public key encryption based on lattice cryptography to encrypt the encoded variables carrying the input data for the function located in a remote server to process. Afterwards, this input data is transmitted over the public network domain and received by the remote server which has the function which conducts the user's desire. The lattice-based cryptographic scheme to initially encrypt the data is a mathematical approach consisting of a matrix-scalar multiplication. Afterwards, the product of the prior operation is added to a variable which is known as noise. The results of the encrypted outputs from the lattice cryptographic scheme are considered to be partially homomorphic results, and it is not until after the server encrypts the output of the computation that the variable achieves the estate of fully homomorphic encryption. It is important to note that the function hosted in the server which has the ability to perform the computation in unencrypted data, is capable of accomplish its goals due to the application of a self-resolving circuit. In other words, the fully homomorphic encryption is accomplished because the design of the function was created with the intent to allow the circuit to be able to formulate or achieve its own decryption. In a simple practical implementation description, imagine that a random person is trying to multiply two extremely big numbers from his mobile phone. In this scenario, the phone performing the calculation will require a great number of resources, from the phone to include CPU and RAM to achieve this calculation. Therefore, leaving the phone completely useless and unresponsive while it is conducting the resource intensive computation. For this reason, it is better not to conduct this computation on the phone because its resources are limited. Hence, it is more viable to outsource the computation to an external server which does have the resources to complete the computations. The implementation of this system is key in this scenario, especially if the data that needs to be transmitted to be operated on is sensitive and its disclosure is not possible. Due to this, the focused target platforms where this solution can be implemented are basically the same platforms which is used to better explain the how system works. This system as referred to by Mayor is known as the client-server system where the client is a weak electronic device incapable of computing a resource intensive operation, and therefore it outsources the task to a remote server which is a powerful computing device with the ability to perform the computations effortlessly.

Underlying mathematical hard problem

Gentry simply defines the system by expressing that "homomorphic encryption is basically a way of delegating a processing of your data without giving away access to it" (Gentry 00:48). Based on this definition, it can be automatically inferred that this system has various segments of applications across different professional departments and fields. Although, the easiest manner to explain the cryptosystem is in a thin-client and server approach, it doesn't necessarily mean that this is the only application of the system. As a matter of fact, according to Player, different entities from different backgrounds are conducting research on the technology to be able to shape it to their own needs. Some of the entities include the government which one of its goal it to allow for secure voting through this scheme. Another popular area where this technology is being researched is in medicine, where there are hopes to implement this technology to provide testing results to patients just to mention two important segments of research. Referring to this information, it is easy to see why homomorphic cryptography is well-liked within the industry and why there is so much focus on the continuance of development of the cryptosystem. Moreover, another big factor which also plays an important role in the advancement of the system is the threat of the emerging sector of quantum computers. It is believed that quantum computing commercialization will be achieve within the next two decades and therefore it is necessary to start standardize the security transmission and communications protocols to ensure that they are resilient against attacks from the quantum security threat. In an attempt to formally start the standardization process, the national institute of standards and technology (NIST) started a campaign giving the public the option to submit proposals for the standardization of the scheme. According to Player NIST "had 69 submission which they considered complete and proper … and of these 23 are in the lattice space which are based on the elderly problem and a related problem called the entry problem" (Player 19:33-19:44). From this information, it can be seen that from all the submissions for the homomorphic encryption scheme that were considered complete and accepted, close to one-third of them adopt a lattice based mathematical approach for encryption. This is due to some special characteristics which are found in the lattice-based cryptographic methodology. One of the lattices based mathematical approach emphasizes in the application of the learning with error (LWE) assumption which consists of adding noise to the product of the secret and public key matrixes. By adding the noise to the product of the encryption and decryption keys, the mathematical problem becomes hard and difficult to resolve. This is due to the fact that figuring out the product of the public and private matrixes before the noise is added is an easy task to do. Nevertheless, when adding the extra factor of noise to the computations, then it becomes very difficult to reverse back to the original matrixes which computed the result. Since the noise factor is an important key in making the overall system a tough mathematical problem, it is also important to emphasize that by adding the noise to final product, a slight distortion of the true value occurs. Moreover, this distortion continues to grow as the noise from other processes are added to the final solution. Because of this, it is necessary to have a mechanism in place which helps to reduce the overall noise of the transaction. This is extremely important since if the noise grows too big then, the fully homomorphic cyphertext would become too noise making it impossible to decipher the encrypted text. It is because of the capacity to conduct noise reduction through the homomorphic evaluation process, which allows the lattice based cryptographic approach to be successful and a secure transmission protocol. It is important to note as well that the learning with errors assumption is the only scheme implemented with lattice-based approaches. As a matter of fact, another popular tactic consists of using NTRUE problems which are also a public key cryptographic system like the learning with errors problem, which is based on the hardness of a certain mathematical problem involving special point in a lattice. It is necessary to note that since the introduction of the cryptosystem, there has been major developments and methods to implement homomorphic cryptographic systems. As it can be seen from the NIST standardization submissions, only about one-third of all accepted proposals implement a lattice-based approach. With this knowledge present, it can be seen that although a lattice-based cryptosystem is the most popular choice when it comes to building homomorphic mathematical structures, it is not the only option on table in order to achieve a homomorphic structure.

Required Computations

Since there are multiple versions with different approaches of the homomorphic crypto system, for the sake of simplicity, the explanation will focus on Gantry's general scheme of utilizing LWE lattice cryptography. Moreover, another factor which makes Gantry's methodology of homomorphic encryption challenging in providing the total number of computations required for the system, is that none of the versions are standardized, it is extremely difficult to calculate the total numbers of computations in the overall system because it utilizes matrices to create the public and private key used for encrypting and decrypting the data. Moreover, this approach also represents the values of the variables in which the homomorphic evaluation is going to be conducted as matrices as well. Since there is no official nor formal standardization of the protocol as of yet, there is not a defined set of dimensions for each of the different matrices implemented in the scheme, then it is practically impossible to obtain a consensus on the overall required computations. This is because as the matrices employed in the process increases in size, also does the number of computations that need to be completed. In other words, the required computations to complete the whole scheme will change drastically based on the dimensions of these matrices. In its most basic break down, the full system can be divided into three major segments. In each of these segments, a number of mathematical operations need to be conducted in order to acquire the overall encryption of the data. On the first stage in the is the creation of the public and private keys. The method to create the keys involve one simple addition and one multiplication. Nevertheless, the total number of multiplications and additions in the creation of the key will be ultimately dependent on the number of elements in the matrices. Once the keys have been created, the process moves to the next phase which involves encoding the data. In the encoding phase of the process, the required computations to accomplish the encryption consist of two matrix multiplications and three matrix additions. Once again, the exact number of multiplications and addition will be dependent on the number of elements of the matrices. After the encoding phase is completed, the data is encrypted with a lattice-based approach and it is ready to be sent to the server conducting the homomorphic evaluation. Once the server receives the encrypted data, it runs it through a special circuit which basically computes the desired operation. The resulting variable from the server is a fully homomorphic encrypted variable which now possess and bootstrap mechanism. This bootstrap mechanism is very important since it contains a self-decrypting circuit which will decipher the contents of the output from the server. Once this process has been completed, the results are sent back to the requestor for decryption. The final stage of the process, includes decoding the cyphertext which was returned by the server in order to retrieve the information. In this final phase, there exist one multiplication and one subtraction of the matrices in order to decrypt the results. Of course, it is necessary to mention that the total number of actual multiplications and subtractions will be dependent on the number of elements of each matrix. To summarize the required computations from running the system, it can be seen the full encryption scheme has four matrix multiplications with four matrix additions and one matrix subtraction. As it can be seen, there are not many overall operations to be computed when implementing the scheme viewed from a matrix perspective. Nevertheless, the number of operations increases drastically as the number of elements increases in each matrix. When this protocol was built, compiled and executed in SageMath to test the run time of its implementation, it took the computer 0.391 milliseconds in order to go through the whole process of generating the keys, encoding the cyphertext and decoding it. It is important to mention that for the sake of this trial, small matrices were implemented. This result is based on an A parameter consisting of a four-by-four matrix, a S parameter consisting of a one by four matrix, and an E parameter of a one by four matrix as well. The computer system conducting the calculations consisted on a HP OMEN desktop with an Intel quadcore I7-7700 CPU with a 3.60GHz frequency per core, 16 GB of ram, and a NVIDIA GeForce GTX 1070 video card with a ram capacity of 1,048,576 bytes.

Conclusions

Homomorphic cryptography continues to be a fast-developing segment in the cryptography field. As of today, NIST continues its quest to formally standardize the protocol. After his PhD thesis introduction, Gantry has continued to improve on the cryptosystem and has work with many other contributors in order increase its efficiency and security. The protocol itself has many important features which makes its implementation very attractive in the sector. Industries like medicine, government, and science can easily find many situations in which implementing this cybersecurity scheme in the not only practical, but also necessary. With this scheme, Gantry has introduced a system that is considered the "holy-grail" of cryptography, and he continues to revolutionize the industry with his advancements in homomorphism cryptography. Moreover, as advancements in the field continue to occur in an expedited manner, this cryptosystem continues to solidify its place in the cryptography spectrum. Key features like data privacy and computational outsourcing makes homomorphic encryption a very appealing solution to implement. Especially in current times, when a lot of the electronic devices use to keep everyone connected will require to outsource the computation since they do not have the computational resources to compute it itself. Nevertheless, this system goes beyond computations field and to only be implemented in this of client-server scenario. Some of the other applications of homomorphic cryptography include genomics, health, national security, data science, satellite communication, machine learning, and education just to mention a few. Based on this list of entities which have found some application of the protocol within their domain, it can be understood that as the system continues to evolve, the number of applications in which this cryptosystem can be used will also increase. to finalize, it is important to note that since the inception of the system, many resources have been made available to the public to assist in the further development of the system. Libraries like MS SEAL which are open source continue to also make it easier for the industry to quickly advance in the sector. Due to this, homomorphic encryption is foreseen to be used in many different aspects for many different purposes. With the mention of this, it is easy to conclude that this cryptosystem is here to stay and is not going anywhere any time soon.

References

Gentry, Craig. *A FULLY HOMOMORPHIC ENCRYPTION SCHEME*. 2009. PhD dissertation

"Homomorphic Encryption." Wikipedia, Wikimedia Foundation, 6 Nov, 2020, https://en.wikipedia.org/wiki/Homomorphic_encryption

"Fully Homomorphic Encryption" YouTube, uploaded by Simons Institute, 15 Jul 2015, Fully Homomorphic Encryption - YouTube

"Security and encoding in Fully Homomorphic Encryption: Rachel Player, Sorbonne Université", YouTube, uploaded by The Alan Touring Institute, 16 Aug 2018, Security and encoding in Fully Homomorphic Encryption: Rachel Player, Sorbonne Université - YouTube

"Winter School on Cryptography: Fully Homomorphic Encryption - Craig Gentry" YouTube, uploaded by Bar-Ilan University, 31 May 2012, Winter School on Cryptography: Fully Homomorphic Encryption - Craig Gentry - YouTube