Handling security issues by using homomorphic encryption in multi-cloud environment

Yulliwas Ameur, Samia Bouzefrane, Thinh Le Vinh

Handling security issues by using homomorphic encryption in multi-cloud environment

Yulliwas Ameur*[a], Samia Bouzefrane[a], Le Vinh Thinh[b]

[a] *CEDRIC Lab, Conservatoire National des Arts et Métiers (Cnam), Paris, France*
[b] *Faculty of Information Technology, HCMUTE, Ho Chi Minh, Vietnam*

Abstract

Taking advantage of the high performance and powerful data processing capabilities of cloud computing technology, externalizing data to the cloud platform is considered as an inevitable trend in the digital field today. However, ensuring the security and privacy of data remains a major challenge. To overcome this drawback, a multi-cloud platform is proposed to improve privacy and high availability of data. A multi-cloud platform that integrates public, private, and managed clouds with a single user interface. Cloud-hosted data is distributed among different data centers in a multi-cloud environment based on cloud reliability and data sensitivity. In terms of security, current encryption algorithms are considered to be very efficient, but it requires a lot of resources to handle this, which is expensive and time consuming. In addition, they also make the data impossible to process without first decoding. To be specific, traditional public key encryption requires data to be decrypted before it can be analyzed or manipulated. In contrast, homomorphic encryption is an encryption method that allows data to be encrypted while it is being processed and manipulated. It allows user or a third party, which can be cloud provider, to apply functions on encrypted data without revealing the data's values. In this paper, we explore existing multi-cloud-based security solutions using homomorphic encryption to identify open issues and opportunities for further enhancement.

Keywords: homomorphic encryption ; multi cloud computing ; security ; privacy; OpenEHR ; electronic medical records

1. Introduction

Cloud computing and related technologies are currently attracting a lot of attention from either research or industry. Michael Armbrust et al [1] define the cloud as "the long-held dream of computing as a utility". The National Institute of Standards and Technology (NIST) gives another definition [2] "cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." The cloud computing paradigm can be described in simple terms as "everything as a service", where data is accessible over the Internet. The most popular ones are Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Software as a Service (SaaS). The advantages of cloud computing are undeniable,

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: yulliwas.ameur@lecnam.net

such as lower computing costs, instant software updates, reduced software costs, and unlimited storage. The multi-cloud service is the next generation in cloud computing's development. The necessity to integrate clouds for enhanced processing and storage capability has become an increasingly relevant subject for IT experts as resource requirements seem to be increasing inexorably. Concerns regarding vendor dependency and cloud failure have also been highlighted, both of which might be mitigated by switching to a multi-cloud environment [3]. A multi-cloud approach is a cloud storage architecture that creates a virtual cloud storage system by combining several cloud storage providers. The data to be saved is divided into separate blocks and redundantly distributed to numerous cloud storage providers. However, collaboration with multiple clouds raises security concerns such as increased attacks, loss of control over data and data privacy issues. To avoid this situation and protect data privacy in multi-cloud environments, one possible solution is to use fully homomorphic encryption (FHE) to ensure privacy. The goal behind FHE is to allow anyone to use encrypted data to perform useful operations without accessing the encryption key. In particular, this concept has applications to improve cloud computing security. In case, a user wants to store sensitive, encrypted data in the cloud but does not trust her cloud provider or is at risk of an instruder breaking into her cloud account or application, FHE provides a way to pull, search, and manipulate data without having to allow the cloud service provider access to the data. Generally, homomorphic encryption is an encryption scheme that allows computations on encrypted data and obtains an encrypted result when decrypted produces the same result of computations on the original data. The aim of this work is to offer an architectural framework to help in securing data sharing processes in a multi-cloud context. To be specific, the paper highlights the ways of achieving secure data sharing through the application of homomorphic encryption. In the next section, we will survey the state of the art on homomorphic encryption and we will explore what are the main challenges to tackle in real applications for multi-cloud architecture. Section 3 provides a detailed description of multi-cloud computing privacy challenges using homomorphic encryption. Introducing the main contribution that describes a new architecture proposed for the multi-cloud electronic medical records, can be found in Sect. 4. Finally, conclusions are summarized in Sect. 5.

2. Related works

To enhance security and privacy in the cloud environment, many studies have tried to use different encryption methods to develop the most optimal solutions. Cloud computing security consists of two areas: security issues encountered by cloud providers (organizations that provide software, platforms or infrastructure) and security issues faced by customers. Innovations and new technologies are fully applied in each part. Therefore, in this section, we conduct a review of the state-of-the-art of different forms of cloud security and try to find out what advantages and insufficiencies exist in current approaches.

In [4], the authors brings a solution for the K-nearest neighbors (k-NN) algorithm with a homomorphic encryption scheme (called TFHE). The proposed solution addresses all stages of k-NN algorithm with fully encrypted data, including the majority vote for the class-label assignment. Unlike existing techniques, the solution does not require intermediate interactions between the server and the client.

In MCC context, the authors [5] discuss the security of mobile multi-cloud computing (MMC) and the advantages for the mobile user(s). They highlight security issues in Mobile Cloud Computing and the main reason to move to multi-cloud. Similary, the light token has been introduced in [6] to secure the mobile user in MCC environment. In this approach, the authors introduced a novel attestation schema based on the existing attestation mechanism as well as traditional enscryptions. Although the stability of the algorithm has been verified, this approach was based on trusted parties, which makes it difficult to control data and privacy. With other approaches, Fabian et al. [7] suggested an architecture for exchanging health care information using Attribute Based Encryption and cryptographic secret sharing. This technique does not ensure the data integrity or efficiency of the full process, which includes uploading, file slicing, and group sharing, among other things. In [8], the authors present a security platform that allows user authentication and data encryption. The platform uses the properties of homomorphic encryption to generate a robust electronic signature. Then to improve the authentication mechanism, the verification tasks are distributed over different virtual machines so that an attacker can never recover or intercept passwords or other personal information of the data subject. In the work of Zibouh et al.[9], a multi-cloud architecture has been proposed with fully homomorphic encryption by using the gentry scheme [10] to enhance the performance and the time of data processing. However if the size of the file increases, computation overhead arises.

3. Multi-cloud computing privacy challenges using homomorphic encryption

Multi-cloud has many advantages for the security of user data in cloud computing. Multi-cloud security is one of the concerns that requires a lot of attention. Many researchers and industry professionals debate on security challenges such as isolation management, data exposure and confidentiality, VM security, trust, and special security risks linked to the cooperation between cloud entities. Trust, policy, and privacy, in particular, are key considerations in multi-cloud systems. We will focus on the customer data protection and identity in this proposal. Cloud data privacy is critical because sensitive customer information should not be shared with anyone who is not authorized to access it. When data is stored in many clouds, there should be a method in place to protect data privacy and identification. When the volume of data is extremely sensitive, clients must disguise their identification traits from Cloud computing services in order to maintain anonymity. To protect unwanted access during data transportation and storage in cloud systems, appropriate data encoding techniques should be utilized.

3.1. Literature review of homomorphic encryption

This new encryption paradigm allows any entity (for example, the cloud provider) to operate on private data in encrypted form without ever decrypting it. The goal of Homomorphic Encryption (HE) is to perform operations on the plain text while manipulating only ciphertexts. Usually, we must decrypt them and then apply the desired processing to manipulate encrypted data. For example, one widespread use case is outsourcing healthcare data to cloud computing services for medical analyses.

More formally an ecryption scheme is called homomorphic over an operation $*$ if it supports the following equation:

$$E(m_1) * E(m_2) = E(m_1 * m_2) \tag{1}$$

where E is the encryption algorithm and M is the set of all possible messages.

According to [11] Homomorphic encryption schemes can be standardised by four algorithms:KeyGen,Enc,Dec and Eval. KeyGen generates a pair (public key, private key) for the assymetric configuration and (private key) for the symmetric version. Enc is the encryption algorithm and Dec is the decryption algorithm.

These three algorithms (KeyGen,Enc,Dec) are common to other conventional cryptosystems, otherwise in homomorphic encryption schemes an additional algorithm is needed, the Eval algorithm defined as follows:

$$Eval(f, C_1, C_2) = f(M_1, M_2) \text{ where } Dec(C_1) = M_1 \text{ and } Dec(C_2) = M_2 \tag{2}$$

For some cryptosystems with algebraic structures, some operations are possible. For example, two RSA ciphertexts can be multiplied to obtain the multiplication of the two corresponding plain texts. We call this property the multiplicative homomorphic property of the ”textbook RSA” cryptosystem. Another operation can also be performed on ciphertexts. For example, in the Paillier cryptosystem [12], we can add two ciphertexts to obtain the addition of the two corresponding plain texts. We call this property the additive property of the ”Paillier” cryptosystem. For example, this can be useful when we are interested in e-voting applications to add encrypted votes without knowing the initial vote.

Rivest, Adelman and Dertouzos first introduced the notion of homomorphic encryption in [13]. Building a cryptosystem with both multiplicative and additive properties was a significant problem in cryptography, until the work of Gentry [14]. Gentry proposed a first Fully homomorphic encryption based on ideal lattices. The HE is categorized depending on the number of mathematical operations performed on the encrypted message as following: **Partially Homomorphic Encryption** (PHE), **Somewhat Homomorphic Encryption** (SHE), and **Fully Homomorphic Encryption** (FHE).

When operations are performed on ciphertexts, the noise increases and too much noise disables accurate decryption. The **bootstrapping** approach is used by FHE systems to get around this as stated in [14]. Bootstrapping decreases the collected noise, allowing further computation. This procedure can be done as many times as necessary to analyze any particular circuit. However, bootstrapping is computationally costly, so many solutions do not employ it in reality. Therefore, we recommend the reader to refer to [15] for more detailed information on the different homomorphic encryption schemes. We have chosen not to describe the entire functioning of cryptosystems due to the lack of space. Also, selecting secure and efficient instantiations of the underlying cryptographic problem is hard for most of encryption and homomorphic schemes. As with Elliptic Curve Cryptography, post-quantum encryption, and many other standardization initiatives, new cryptographic concepts take a few years to achieve general adoption in the industry. FHE is only appropriate for single-user calculations since it needs inputs to be encrypted with the same key. However, there is a variety of scenarios in which users who have uploaded their big data stores to the cloud in encrypted form decide to calculate a joint function of their combined data. They may, for example, want the cloud to calculate joint statistical information on their databases, discover common files in their collections, or operate a computational agent to make a decision based on their pooled data (without disclosing anything but the final judgment).

3.2. Multi-key Homomorphic encryption

López-Alt et al. [16] proposed the first multi-key homomorphic encryption,a system to provide a homomorphic evaluation on ciphertexts encrypted with various keys. Unlike general HE schemes, this kind of HE scheme eliminates the necessity for a key setup step prior to any computation to build a joint key from individual keys. Instead, a cloud evaluator may dynamically convert ciphertexts from encryption using individual keys for encryption using the concatenation of individual users' keys.

We have proposed a solution using the following scheme [17], named MK-TFHE scheme, a first implementation in the literature to implement an MKHE scheme.which is defined by seven probabilistic polynomial-time (PPT) algorithms:

- pp ← MK-TFHE.Setup(1^k): This algorithm outputs public parameters pp given security parameter k
- sk ← MK-TFHE.KeyGen(pp): This algorithm randomly generates secret key sk given the public parameters pp.
- ct ← MK-TFHE.Encrypt(m,sk): This randomised algorithm encrypts message m with secret key sk and outputs ciphertext ct=(b,a).
- (ct^*, T^*) ← MK-TFHE.Pre-process ($\overline{ct}$=(b,a_1,...,a_{ki}),T=id_1,...,id_k): This algorithm basically extends the input ciphertext with additional 0's in order to be able to perform the homomorphic operation over all the underlying k keys.
- ct^* ← MK-TFHE.Eval(C,ct^*): This algorithm evaluates circuit C over the l ciphertexts ecrypted with multiple keys.
- u_i ← MK-TFHE.PartialDecrypt(a_i^*, sk_{id_i}): This algorithm takes as input a_i^* from ciphertext ct^* corresponding to the party holding secret key sk_{id_i} and outputs a partially decrypted information u_i
- m' ← MK-TFHE.Merge($b, u_1, ..., u_k$): This algorithm takes as input all the partial decryptions derived from a ciphertext ct^* and outputs the final plaintext m'

4. Our contribution

To further enhance data privacy and reduce the amount of calculations, we proposed more secure system models by adding more clouds.

Many challenges need to be tackled to apply homomorphic encryption in multi-cloud real-world applications. We have identified the multi-key homomorphic encryption [18] or the multiparty extenstions "Threshold setup" of the library OpenFHE [19] for BGV, BFV, and CKKS schemes, to use them in a multi cloud setup.

We introduce and describe a new architecture proposed for multi-cloud electronic medical records as in Fig. 2. This model uses homomorphic encryption algorithms to ensure individual privacy in network and multi-cloud environments. The advantage of our proposal is that it is based solely on OpenFHE, a reference open-source library in

the field of homomorphic encryption, and can be easily configured by neophytes. We performed simple homomorphic operations to prove the feasibility as it is depicted in the next sub-sections.

4.1. Experimental evaluation

We make some experiments in order to evaluate the performance and the usability aspects on a variety of applications of multi-key and multiparty extenstions "Threshold" of homomorphic encryption. In the multi-cloud environment, we use OpenFHE [19] for Homomorphic Encryption and DepSky as a multi-cloud platform.

4.1.1. OpenFHE: Open-Source Fully Homomorphic Encryption Library

We use OpenFHE[19] to implement the homomorphic encryption scheme, a new open-source FHE software library that integrates a variety of innovative design concepts from previous FHE libraries such as PALISADE, HElib, and HEAAN. OpenFHE supports various FHE schemes and hardware acceleration backends using a standard Hardware Abstraction Layer (HAL). OpenFHE supports both user-friendly and compiler-friendly modes, with the library automatically performing all maintenance operations such as modulus switching, key switching, and bootstrapping. We choose to use a multi-key homomorphic encryption scheme [17], a cryptosystem that allows us to evaluate an arithmetic circuit on ciphertexts, possibly encrypted under different keys.

4.1.2. DepSky: Multi cloud computing platform

We experiment with our proposal by using DepSky[20] with local storage, a multi-cloud platform that enhances the integrity, confidentiality, and accessibility of data stored in the cloud. This is accomplished by creating a cloud-of-clouds by encrypting, enclosing, and replicating all the data across a number of separate clouds. This architecture addresses the single cloud's limitations by replicating all of the data in a set, the availability issue of clouds and as a result the data can be retrieved correctly even if some of the clouds corrupt or lost data. It addresses the loss and corruption of data issue by using Byzantine fault-tolerance replication to store data in multi-clouds. It addresses the loss of confidentiality issue by employing a secret sharing schema and erasure codes to ensure that all data that will be stored in a multi-cloud environment are encrypted.

4.1.3. The Health-Care Use-Case

Nowadays, the application of multi-cloud environment is more and more widespread. Multi-cloud environment is applied in many different aspects such as information communication, vehicle systems and medical health systems. While people are paying more and more attention to their medical health, their medical data considered as sensitive must be handled in a secure way like shown in Fig.1.

Several health-care activities using EHRs (Electronic Health Records) are an attractive use-case for multi-provider cloud in the healthcare industry: data access by the patients, prescription management application for doctors or other institutions, and assisted surgery.

The protection of personal health data is a major issue. Indeed, these data can lead to the covetousness of malicious parties with the aim of generating profits [21]. The three fundamental and main goals of security are: confidentiality, integrity and availability.

- Confidentiality: Only an authorized person should have access to the information.
- Integrity: Information should be correct, and an unauthorized person should not alter it.
- Availability: Information should be accessible, available, and usable at any time but only by an authorized entity.

4.1.4. Architecture Model

Because many applications may use sensitive data that are distributed over multiple clouds, in our proposal, we propose a new architecture model that is based on homomorphic encryption in a multi-cloud environment. By combining these two paradigms, we design an architecture model which can ensure the security and the privacy of the users. The proposed architecture for our proposal is summarized in Fig. 2.

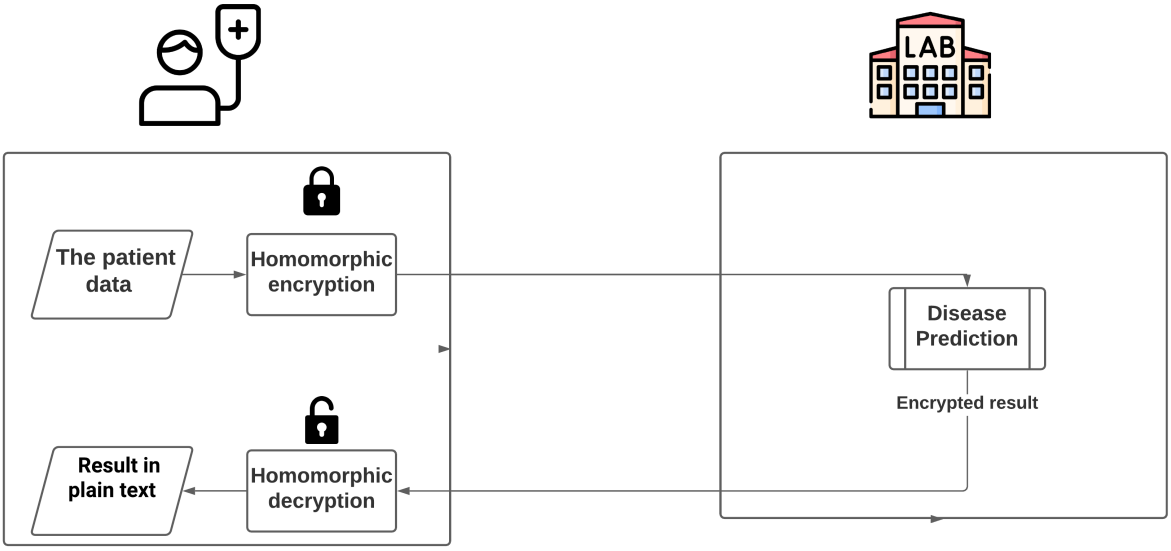

Fig. 1: ”Healthcare monitoring with a single cloud.”

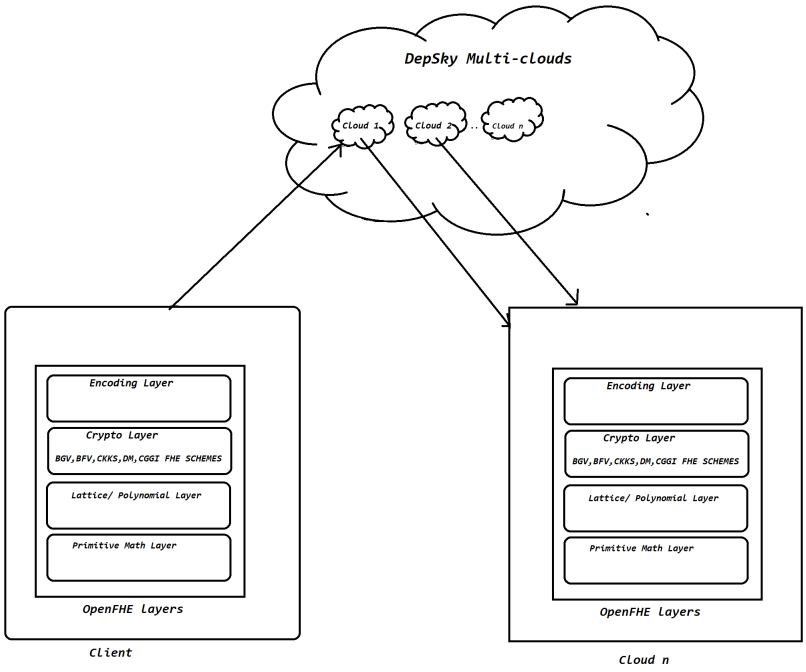

Fig. 2: ”diagram showing how to manipulate encrypted data on a multi-cloud platform DepSky. On the left, the client encrypts the data before sending it to the cloud, on the right, the cloud service can process on it.”

4.2. Detailed experimental results

Our solution has been implemented using OpenFHE and DepSky, and tested on Linux Ubuntu 64-bit machine with i7-7700 CPU 3.60GHz with four clouds.

The advantage of our architecture is that it is possible to choose among several homomorphic encryption schemes depending on the needs of homomorphic operations. We did our simulation using four local clouds, nevertheless, it is easy to deploy by choosing a commercial cloud like Amazon S3, Google Storage, RackSpace Files or Windows Azure Storage.

For security parameters, we have used the recommended parameters [17].

Table 1 describes the parameter used for our experimentation.

LWE				RLWE	
n	α	B	d	N	β
560	3.05*10^{-5}	2^2	8	1024	3.72*10^{-9}

Table 1: Recommended parameter settings for our MKTFHE scheme: n, α and N, β denote the dimension and the standard deviations for LWE and RLWE ciphertexts to achieve at least 110-bit security level

To show the feasibility of a homomorphic computation on a multi-cloud service we have applied simple operations such as an addition or a multiplication between two integers, the results obtained are presented in Table 2.

The results obtained are promising compared to those obtained with a mono-cloud service, because indeed the multi-cloud architecture allows to make parallel calculations over multiple data encrypted with multiple keys, with an additional negligible cost when we perform the decryption operation using the concatenation of individual user's keys.

It would now be interesting to test more complex operations such as applying machine learning algorithms on health data by using homomorphic encryption [22].

5. Conclusion

Despite the benefits of cloud applications for healthcare, cloud security challenges must be addressed. In this paper, we introduce and describe a proposed new architecture for Electronic Health Records with Multi-Clouds. This model will ensure the privacy of people in a network and multi-cloud environment using a multi-key homomorphic encryption algorithm. The advantage of our proposal is that it is based only on an opensource library "OpenFHE" that is a reference in the field of homomorphic encryption, and the configuration is easy for a non-expert. We have performed simple homomorphic operations to prove the feasibility. However, it would be interesting to explore other types of more complex operations, such as machine learning algorithms to know the contribution of the multi-cloud for the acceleration of the computation time.

Number of clouds	**time of one addition**	**Time of a multiplication**	**Blind rotation key**	**Key-switching key**
1	63 μs	1200 μs	0.62 MB	70.3 MB
2	48 μs	580 μs	0.82 MB	79.1 MB
4	25 μs	320 μs	1.03 MB	95.1 MB
8	15 μs	203 μs	1.33 MB	100.3 MB
16	9 μs	124 μs	1.62 MB	120.2 MB

Table 2: Results obtained of the calculation time of an addition and a multiplication by varying the number of clouds, with the size of the blind rotation key and the key-switching key

References

[1] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy H Katz, Andrew Konwinski, Gunho Lee, David A Patterson, Ariel Rabkin, Ion Stoica, et al. Above the clouds: A berkeley view of cloud computing. Technical report, Technical Report UCB/EECS-2009-28, EECS Department, University of California . . . , 2009.

[2] Peter M. Mell and Timothy Grance. Sp 800-145. the nist definition of cloud computing. Technical report, Gaithersburg, MD, USA, 2011.

[3] Hussam Abu-Libdeh, Lonnie Princehouse, and Hakim Weatherspoon. Racs: a case for cloud storage diversity. In *Proceedings of the 1st ACM symposium on Cloud computing*, pages 229–240, 2010.

[4] Yulliwas Ameur, Rezak Aziz, Vincent Audigier, and Samia Bouzefrane. Secure and non-interactive-nn classifier using symmetric fully homomorphic encryption. In *International Conference on Privacy in Statistical Databases*, pages 142–154. Springer, 2022.

[5] Maya Louk and Hyotaek Lim. Homomorphic encryption in mobile multi cloud computing. In *2015 International Conference on Information Networking (ICOIN)*, pages 493–497, 2015.

[6] Thinh le, Herve Cagnon, Samia Bouzefrane, and Soumya Banerjee. Property based token attestation in mobile computing. *Concurrency and Computation: Practice and Experience*, 32, 10 2017.

[7] Benjamin Fabian, Tatiana Ermakova, and Philipp Junghanns. Collaborative and secure sharing of healthcare data in multi-clouds. *Information Systems*, 48:132–150, 2015.

[8] Karim Zkik, Ghizlane Orhanou, and Said El Hajji. Secure scheme on mobile multi cloud computing based on homomorphic encryption. In *2016 International Conference on Engineering MIS (ICEMIS)*, pages 1–6, 2016.

[9] O. Zibouh, A. Dalli, and Hicham Drissi. Cloud computing security through parallelizing fully homomorphic encryption applied to multi-cloud approach. 87:300–307, 05 2016.

[10] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 169–178, New York, NY, USA, 2009. Association for Computing Machinery.

[11] Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai, and Vinod Vaikuntanathan. Homomorphic encryption security standard. Technical report, HomomorphicEncryption.org, Toronto, Canada, November 2018.

[12] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*, pages 223–238. Springer, 1999.

[13] R L Rivest, L Adleman, and M L Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Computation, Academia Press*, pages 169–179, 1978.

[14] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178. ACM, 2009.

[15] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Comput. Surv.*, 51(4), jul 2018.

[16] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1219–1234, 2012.

[17] Hao Chen, Ilaria Chillotti, and Yongsoo Song. Multi-key homomorphic encryption from tfhe. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019*, pages 446–472, Cham, 2019. Springer International Publishing.

[18] Asma Aloufi, Peizhao Hu, Yongsoo Song, and Kristin Lauter. Computing blindfolded on data homomorphically encrypted under multiple keys: A survey. *ACM Comput. Surv.*, 54(9), oct 2021.

[19] Ahmad Al Badawi, Jack Bates, Flavio Bergamaschi, David Bruce Cousins, Saroja Erabelli, Nicholas Genise, Shai Halevi, Hamish Hunt, Andrey Kim, Yongwoo Lee, Zeyu Liu, Daniele Micciancio, Ian Quah, Yuriy Polyakov, Saraswathy R.V., Kurt Rohloff, Jonathan Saylor, Dmitriy Suponitsky, Matthew Triplett, Vinod Vaikuntanathan, and Vincent Zucca. Openfhe: Open-source fully homomorphic encryption library. Cryptology ePrint Archive, Paper 2022/915, 2022. https://eprint.iacr.org/2022/915.

[20] bessani Alysson Bessani Ricardo Mendes. "dependable and secure storage in a cloud-of-clouds", 2016.

[21] H. Pussewalage and V. Oleshchuk. A patient-centric attribute based access control scheme for secure sharing of personal health records using cloud computing. In *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*, pages 46–53, Los Alamitos, CA, USA, nov 2016. IEEE Computer Society.

[22] Yulliwas Ameur, Samia Bouzefrane, and Vincent Audigier. Application of homomorphic encryption in machine learning. In *Emerging Trends in Cybersecurity Applications*, pages 391–410. Springer, 2023.