RESEARCH REVIEW JOURNALS
Volume-04
Issue-02
February -2019

ISSN: 2455-3085 (Online)
RESEARCH REVIEW International Journal of Multidisciplinary
www.rrjournals.com [UGC Listed Journal]

A state of art review on homomorphic schemes in cloud computation

*1Sindhu V & 2Dr. Rajeswari Mukesh

1Research Scholar, Bharathiyar University, Coimbatore, Tamilnadu (India)
2Head of the Department, Department of CSE, Hindustan University, Chennai, Tamilnadu (India)

ARTICLE DETAILS

Article History
Published Online: 10 February 2019

Keywords
Homomorphic encryption, cloud computing, privacy.

***Corresponding Author**
Email: sindhujan28[at]gmail.com

ABSTRACT

Cloud service providers play a vital role in data storage and data processing by providing software, platform and infrastructure as service. The third party cloud vendors allow several clients to share the resources. The resource sharing is a threat for the privacy of the user data. Other than the cotenants the parties involved in data access also affect the privacy of user information. Researches introduced the homomorphic algorithms that do not disclose the encrypted data during processing it. In this paper, we discuss the different schemes proposed to implement homomorphism and the factors that still to be addressed in implementing homomorphism in real time.

1. Introduction

The revolution in computing devices and network resulted in increased data repositories. IDC predicted that the data growth rate will increase from 1.8 zettabytes8 (ZB) in 2011 to over 7 ZB by 2015. Also IDC foresee that the Global Datasphere will grow from 33 Zettabytes (ZB) in 2018 to 175 ZB by 2025.[2]. The increase in public and open source cloud service providers attract users towards public storage medium served through cloud applications [eg: Dropbox, icloud, Onedrive, Googledrive, Amazondrive].

As data rules the business, data in any format is useful for the industry to make managerial decisions. It could be structured, semi-structured or unstructured. Also the data could be live or from a repository. The data created through any device or process goes through a data lifecycle. The data lifecycle starts with formation of data and ends with archive followed by destroying. The intermediate processing involved in data are depicted in figure 1 that shows the data flow in cloud computing. Security breach occurs in all the stages from transfer over network till destroy[13]. The level of threat is high when the data is kept in third party public cloud servers.

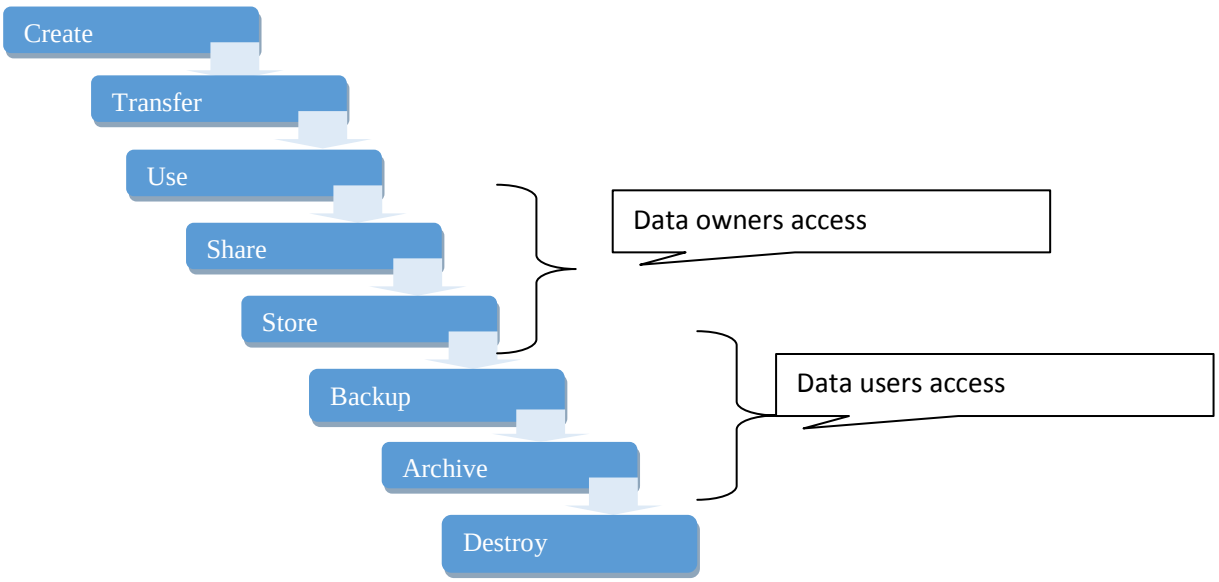

Figure 1: Data Lifecycle

The cloud user risks by losing personnel, logical and physical control on data. The data breaches include encryption key loss, vulnerabilities in virtualization, SaaS, data leakage, interception, economic and distributed denial of service[21]. Social, national, health, financial data suffer authorization, authentication and access control issues.

When the private and community cloud is under third party control data suffers out of potential loss of intellectual property and business secrets. The users might get locked with the vendor and find intricacy in migrating data to new vendor. The globally distributed cloud data result in country jurisdiction issues and privacy issues. The users provide personal information and allow access to the device storage to cloud application that lead to loss of privacy.

The progress in cloud architecture provides enormous storage to devices that handle data virtually through communication network. The encryption techniques are used to enforce security to data in network and storage. The unknown resources under the control of the third party service providers are involved in data computations. The confidentiality of data residing in unsecured cloud could be ensured by storing encrypted data. This protects the data from cotenants residing in the same cloud with data leakage during computations(figure 2). Computation on encrypted data should be implemented to ensure complete data protection from cotenants and service providers. The encrypted computation results are decrypted only at the client end is enabled though homomorphic encryption [23]. This plays a major role in providing privacy to sensitive data.

Globalization and healthcare network go ahead to store confidential data on community cloud or public cloud that are weak to leak the information. The data is secured through various encryption algorithms that protect it from hackers or intruders but not from the cloud service providers. The sensitive records should not be visible to anyone without the data owner's permission including the cloud server applications.

Homomorphism provides the privacy in which the cipher text transferred to the cloud is processed in encrypted form and decrypted only at the user end. This paper presents the state of art in homomorphic algorithms used in cloud for managing privacy of data for both the data owners and data users with plain/encrypted request to the data. The next sections of the paper includes the evolution of homomorphic algorithms, related study on the proposed methods to implement homomorphism in cloud, analysis of the key factors that influence the practicality of homomorphism in cloud and suggest the mitigations.

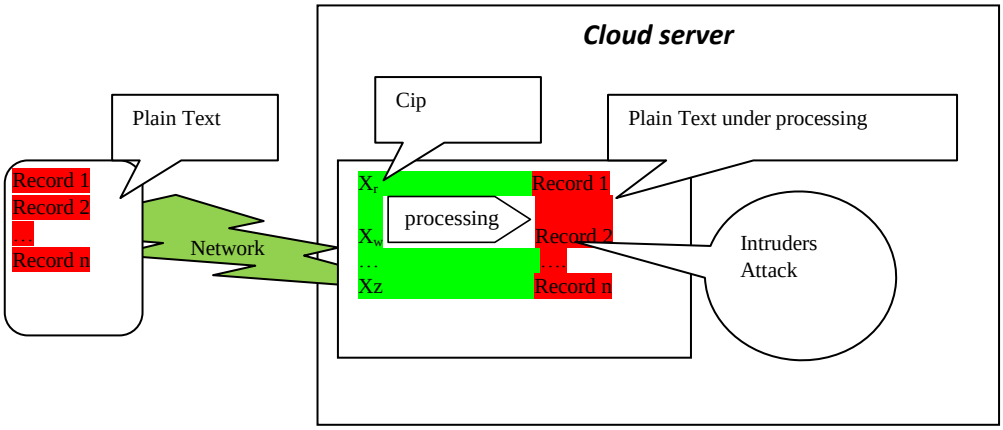

Figure 2: Attack On Data In Processing

2. Evolution of Homomorphic Encryption

Homomorphism is a map maintaining all the algebraic structures between domain and range of an algebraic set in abstract algebra. The input for the map is from the set of domain and the output is the range. Generally encryption techniques cannot process the cipher text in which the operations are performed only on decrypted plain text. This results in compromising privacy for execution. Furthermore un-trusted servers, service providers and cloud operators can keep identifying information of users even after the users end the relationship with the services [McMillan 2013].Homomorphic encryption allows processing data in encrypted form that ensures the user information is not leaked in the server end. The functioning of homomorphic operation is illustrated as if op_1 and op_2 being two operands stored in cloud in encrypted form using the encryption key k as k(op_1)and k(op_2)then homomorphism could compute k($op_1 + op_2$) and the resultop_1+op_2 is decrypted only at the client machine. This way the data leakage is completely removed in the third party cloud storage.

Homomorphic encryption techniques are widely used in cloud computing which involves untrusted servers and cotenants. The scenarios benefit out of encrypted processing are outsourcing storage and computations, private queries, two party computations and zero knowledge applications[8].

The Homomorphic encryption is classified as partial, somewhat and fully Homomorphic algorithms based on the operations carried out in cipher text. PHE (partially homomorphic encryption) schemes are homomorphic with respect to only one type of operation: addition or multiplication. SHE (somewhat Homomorphic encryption supports homomorphic operations with additions and multiplications. The disadvantage is that it executes only limited operation. FHE (fully Homomorphic encryption) allows unbounded number of operations.

Table 1Homomorphic Encryption Algorithm Evolution

Scientist	Type	Nature of Operation	Year
Rivest et al	PHE	Multiplicative	1978
Goldwasser-Micali	PHE	Additive	1984
ElGamal	PHE	Multiplicative	1985
Okamoto–Uchiyama	PHE	Additive	1998

Paillier	PHE	Additive	1999
Gentry	FHE	Additive and multiplicative	2009
Marten van Dijk	FHE	Additive and Multiplicative on integer	2010
Ivan Damgard	SHE	Arithmetic	2012
Dan Boneh	SHE	Query Processing	2013

Rivest et al. published “On Data Banks and Privacy Homomorphism” explaining how a small loan company could use a commercial time-sharig service to store and compute on encrypted data in the year 1978[8].The proposal suffered crucial computation overhead and complexity.

ShafiGoldwasser and Silvio Micali et al proposed an asymmetric key encryption algorithm developed by in 1982. Its disadvantage is that the ciphertext is several hundred times larger than the initial plaintext. Yao proposed two party communication protocol to compare the wealth of two rich people masking the exact amount to each other in the year1984[25]. Ciphertext size grows linearly with the computation of every gate in the circuit. ElGamal encryption system is based on the difficulty of calculating discrete logarithms in a finite field published in 1985[24]. The disadvantage of ElGamal encryption scheme is the length of ciphertext which is double the initial text and comparatively longer than Rivest et al homomorphism.

Okamoto and Uchiyama et al developed a cryptosystemin1998.Paillier et al. published “The encryption scheme of Page” the additive homomorphic encryption scheme with public key in 1999. The cryptosystem suffers on the difficulty of factorization of a composite number as a product of two primes. Gentry et al proposed a fully homomorphic encryption scheme in 2009 that performs both additive and multiplicative operations on polynomials, it is slow and can solve only small circuits[22]. Marten van Dijk et al proposed a fully homomorphic scheme that performs additive and multiplicative operations over integers[20].

Ivan Damg˚ard et al [2012] proposed a scheme that performs distributed decryption and handle many values in parallel in one ciphertext. It performs either addition or multiplication [19]. Dan Boneh et al [2013] proposed a somewhat homomorphic scheme that use polynomial encoding of the database to implement conjuction queries. The limitation of the scheme is for every query the server should perform computations on each record in the database which is not suitable for very large databases[18].

3. Literature Review

The cloud data access could be categorized into data owners and data users[fig 1]. The data owners are the one who created the data and has complete access on it. The owners access the live data until it is valid and later it is archived for managerial decisions. The data users are the one who applies aggregative access on the live or archived data. The actual data is not revealed to the data users but they are allowed to manipulate summative operations on it with owner’s /service provider permission.

Youssef Gahi et al[2015] proposed a novel architecture to execute query statements over encrypted data in 2015. The client encrypts the query parameters and sends the request to the server. The server is clueless about the content and its position while performing the operation. The proposed scheme use Gentry’s Homomorphic algorithms[14] that suffers large encrypted data. The disadvantage is the database occupies a very large memory space and the operations are time consuming. Also the mechanism encrypts the content and not the query so the query operations are explicit in the server.

O. Kocabas et al[2015]proposed a encrypted ECG record mechanism in cloud storage. The leakage of health data results in lack of privacy of the patients. So the proposed method encrypts the ECG data and store it in the cloud and decrypts the same only in the doctors mobile device[15][16]. The mechanism suffers the disadvantages of Homomorphic encryption that computes a large cipher text. So the authors present a encrypted data computation using leveled fully homomorphic encryption[17]. The leveled FHE overcomes the limitations of FHE that decrease the number of multiplicative operations to reduce the size of the encrypted text. The limitations are due to the size of the level that increases the storage and expensive multiplication operation.

MekalaBhaskar et al[2017] proposed a scheme in which data is stored in Amazon web service using dynamoDB. The method is implemented using fully Homomorphic encryption in which user computations are performed on the encrypted data in cloud storage. The dynamoDB is manipulated using java code to perform addition and subtraction on encrypted data. The data is not allowed to be stored as heterogeneous text. Also the scheme suffers largest cipher text and limited querying operations[11].

JiafengHua et al[2018] propose an efficient and privacy-preserving medicalprimary diagnosis framework (CAMPS).In this mechanism diagnosis models are stored in encrypted form incloud server and the users can access it using skyline computations without revealing the data using partial decryption with a special fast secure two party vector dominance method. The proposed method use leveled homomorphic scheme[17] and also requires user registration as a proof of identity to run masked queries. But it fails to prevent collusion[5].

Dennis Grishin et al[2018]. The human genome project assembles DNA sequencing of patients and healthy individuals at affordable cost. Tests are performed on personal genome sequencing are referred as genotyping. It is necessary to treat genetic diseases. The majorities of the variants is distributed throughout the genome and remain undiscovered. The blockchain mechanismallows the buyer to execute a contract providing the blockchain address and the content of the

address are the data blocks. The validator node verify the integrity of the requested data and encrypt the data using buyers public key. In privacy perception the buyers define the query and genetic data that is encrypted with collective public key generated by validator node. The encrypted queries execute encrypted data without knowing the actual data using homomorphism. The results are decrypted at data buyer end usingprivate key of the data buyer[3].

J L Raisaro et al[2018] Informatics for Integrating Biology and the Bedside (i2b2)is an open source clinical platform for enabling secondary use of electronic health records (EHR) used at over 150 clinical sites and covering more than 250 million patients' records only in the US. The author proposes a shared key concept to encrypt the data and the query to process the data using Elgamalencryption[24]. The data encryption is implemented using the proxy server and the cloud server in which the partial key is generated using the public key of both the servers and the private key of the remains secret. It also include an investigator to generate the asymmetric keys to encrypt the query to process the encrypted data. The results are re-encrypted using investigators public key so as to decrypt in the user end using investigators private key. It is based on Elgamal additive Homomorphic encryption scheme to get the total number of patients. The proposed scheme ignores malicious adversary[6].

Marwan M et al[2017] propose a Region of Interest encryption using multi-agent system to store medical images in cloud storage. Instead of encrypting the whole image only the required part of the image is stored encrypted in the cloud storage by the multi agent system. Although the Homomorphic schemes provide necessary privacy it creates computational overhead and the encrypted data occupies large space in the third party cloud servers. So the region of interest encryption converts only the partial data of interest to cipher text that provides the advantage of reducing the processing time and less memory space utilization. But what data to be encrypted is based on multi agent'sdecision[10].

DING et al[2017] propose a scheme that employs Homomorphic re-encryption scheme to provide multiple acess control servers(ACS) to execute their encrypted queries on encrypted data stored in untrusted cloud servers. The data requesters send the encrypted query to the cloud server that is computed on encrypted data. To check the confidentiality of the requester the reply is re-encrypted with ACS public key and sent to theuserthrough the ACS. Only the trust worthy requesters could decrypt the data. The multiple ACS involved is believed to be confidential.[7]

SergiuCarpov et al [2016] proposed a fully Homomorphic encryption method to preserve the user's health data using asymmetric key.The user alone can decrypt the data which is sent in encrypted form in the cloud using the same user's public key. The disadvantage is if the data is to the shared with another component then the proposed scheme requires symmetric key transmission using cloud service providers public key encryption.[12].

WEI GUO et al[2018] propose a privacy-preserving online medical pre-diagnosis scheme(POMP), to protect the privacy of the healthcare user from Hospital Service Provider and the cloud service provider using asymmetric keys. As the data is encrypted by the user it provide strict privacy to the user both from the HSP and the CP.[4]. And the curioususer is also not allowed to access the cotenant data in available in data source service provider.

MouhibIbtihal et al[2017] proposed an architecture in to protect images stored in mobile clouds that involves three steps. First image isencrypted and stored in the private cloud. Secondly paillier'shomomophic encryption is applied when it is stored in public cloud. And finally the encrypted image in public cloud is accessed with watermarking to assure the originality and integrity of the image stored[9]. The limitation is a private cloud used to enforce additional encryption to the cloud image data.

M.R.Baharon et al[2018] invented a fully homomorphic scheme to transcode videos using cloud service. Video transcoding is a mechanism that converts video in one format to another. When the video is transcoded using cloud service, it should be uploaded in cloud storage. This upload process allows untrusted service provider and the cotenants to view the video data. To avoid video data leakage, the author initiated a symmetric key fully homomorphic encryption to upload the encrypted video file that is transcoded in encrypted form and decrypted only at the client end. The data is temporarily uploaded only during the time of transcoding so the method overcomes the limitations of symmetric key and large size encrypted file[1].

The various techniques suggest use either full or partial or somewhat homomorphism to protect the data. Almost all them use a public and private key pair to access the data whereas only few concentrates on the secrecy of the data requestor[3,6]. When the query patterns are leaked to the intruders it risks the privacy of the data.

Table 2: Nature of the homomorphic techniques used in cloud computation

Schemes	Algorithm	Nature of Encryption	Query pattern	Multi - servers	Access right
1	FHE	Asymmetric	Plain query	×	owner
2	Leveled FHE	Asymmetric	Plain query	×	owner
3	FHE	Asymmetric	Limited query	×	owner
4	Leveled FHE	Asymmetric	Skyline query	✓	owner/ user
5	FHE	Asymmetric	Encrypted Query	✓	owner/user
6	PHE	Asymmetric	Encrypted Query	✓	Owner/ user

7	FHE	Asymmetric	----	✓	Owner/ user
8	PHE with re-encryption	Asymmetric	Plain query	✓	Owner/ user
9	FHE	Asymmetric	Plain query	×	Owner/ user
10	SHE	Asymmetric	Plain query	×	Owner
11	PHE	Asymmetric	Plain query	✓	Owner
12	FHE	symmetric	NIL	×	Owner

4. Barriers in cloud

The above schemes contribute towards privacy for the data including text, image and video, combining homomorphic algorithms with data storage mechanism. But homomorphism is not practically implemented due to the computation and communication overhead due to noisy data. Additional factors that affect the practicality of homomorphism in cloud are the validity of the asymmetric key generators, the number and type of servers involved in processing the data, the privacy of the request issued to process the data and the type of components that have the rights to access the data.

General drawbacks of homomorphic algorithms: Even homomorphism has evolved in performance it is not suitable in real time because of large noisy data introduced in encryption processing. This also results in long computation time that increases the price of computation involving cloud resources. Also the communication cost increase as the data is transferred in encrypted form. Another problem with homomorphism is it is cannot perform all the operations[25,24,20,8]. Querying with homomorphic algorithms is also complex.

Asymmetric & symmetric key: The private and public key combinations in encryption reduce the speed in computation. Multiple parties participating in data access requires key exchange or should involve third party key generator servers. The symmetric key mechanism is strongest but it requires techniques to secure the confidentiality in key exchange[14,15,17].

Multi-server participation: Cloud services are not restricted to single server. It includes public and private servers that share the computations. It involves access control and key servers that provide authentication for the component accessing the data and generate public private key pairs for them. The limitation with multiple server computation is collusion where there is a possibility of information leakage between the servers[1,3,6,10].

Secured Access: The data stored in cloud is encrypted and processed in encrypted form but the request to the data should also be encrypted to protect the request from the intruders. A plain request/query allows guessing of data in encrypted form[7].

Data Access: There are three categories of data access components: 1. Data owner, 2. Data provider, and 3. Data user. The data owners are the one to whom it belongs to whereas data providers are from where it has been generated like a date of birth of account holder generated for the bank. In which the account holder is the data owner and the bank is the data provider. Next is the data user who generates aggregated information of the encrypted data in cloud. The difficulty lies in authenticating all the data access components with protective access control and key mechanisms[1,7].

In homomorphic encryption the above four factors play a major role in depressing its practicality. So a secured authentication and key server mechanism is required with homomorphism to implement full privacy for the data access from cloud servers.

5. Conclusion

The paper discuss about various proposals for homomorphic processing on accessing the data stored in cloud service providers disks. Even though the actual data is secured under homomorphism, different parties involved in processing have to be controlled using strict authentication. The principle drawback of encrypted processing is the data size. This could be addressed by deduplicating the data. The future work is to design a fully protective authenticated homomorphic technique to cloud server access.

References

1. Baharon, MohdRizuan, et al. "SECURE VIDEO TRANSCODING IN MOBILE CLOUD COMPUTING." *International Journal of Computing* 17.4 (2018): 208-218.
2. David Reinsel , John Gantz and John Rydning, "The Digitization of the World From Edge to Core", An IDC White Paper – #US44413318, Sponsored by November 2018.
3. Grishin, Dennis, et al. "Accelerating Genomic Data Generation and Facilitating Genomic Data Access Using Decentralization, Privacy-Preserving Technologies and Equitable Compensation." *Blockchain in Healthcare Today* (2018).
4. Guo, Wei, et al. "A privacy-preserving online medical prediagnosis scheme for cloud environment." *IEEE Access* 6 (2018): 48946-48957.
5. Hua, Jiafeng, et al. "CAMPS: Efficient and privacy-preserving medical primary diagnosis over outsourced cloud." *Information Sciences* (2018).
6. Raisaro, Jean Louis, et al. "Feasibility of Homomorphic Encryption for Sharing I2B2 Aggregate-Level Data in the Cloud." *AMIA Summits on Translational Science Proceedings*2017 (2018): 176.
7. Ding, Wenxiu, Zheng Yan, and Robert H. Deng. "Encrypted data processing with homomorphic re-encryption." *Information Sciences* 409 (2017): 35-55.

8. Halevi, Shai. "Homomorphic encryption." *Tutorials on the Foundations of Cryptography*.Springer, Cham, 2017.219-276.
9. Ibtihal, Mouhib, and Naanani Hassan. "Homomorphic encryption as a service for outsourced images in mobile cloud computing environment." *International Journal of Cloud Applications and Computing (IJCAC)* 7.2 (2017): 27-40.
10. Marwan, Mbarek, Ali Kartit, and Hassan Ouahmane. "Towards Optimizing the Usability of Homomorphic Encryption in Cloud-Based Medical Image Processing." *International Symposium on Ubiquitous Networking*.Springer, Cham, 2017.
11. MekalaBhaskar, Dr M. Ashok, and M. D. A. Hasan. "Homomorphic Encryption Algorithm used in Multi-Format Data in Collaborated Healthcare Multi cloud Computing." (2017).
12. Carpov, Sergiu, et al. "Practical privacy-preserving medical diagnosis using homomorphic encryption." *Cloud Computing (CLOUD), 2016 IEEE 9th International Conference on*.IEEE, 2016.
13. Bamiah, MervatAdib. *Trusted Cloud Computing Frame Work in Critical Industrial Application*.Diss. UniversitiTeknologi Malaysia, 2015.
14. Gahi, Youssef, MouhcineGuennoun, and Khalil El-Khatib. "A secure database system using homomorphic encryption schemes." *arXiv preprint arXiv:1512.03498* (2015).
15. Kocabas, Ovunc, and TolgaSoyata. "Towards privacy-preserving medical cloud computing using homomorphic encryption." *Enabling Real-Time Mobile Cloud Computing through Emerging Technologies*.IGI Global, 2015.213-246.
16. Kocabas, Ovunc, and TolgaSoyata. "Utilizing homomorphic encryption to implement secure and private medical cloud computing." *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*.IEEE, 2015.
17. Brakerski, Zvika, Craig Gentry, and VinodVaikuntanathan. "(Leveled) fully homomorphic encryption without bootstrapping." *ACM Transactions on Computation Theory (TOCT)* 6.3 (2014): 13.
18. Boneh, Dan, et al. "Private database queries using somewhat homomorphic encryption." *International Conference on Applied Cryptography and Network Security*.Springer, Berlin, Heidelberg, 2013.
19. Damgård, Ivan, et al. "Multiparty computation from somewhat homomorphic encryption." *Advances in Cryptology–CRYPTO 2012*.Springer, Berlin, Heidelberg, 2012.643-662.
20. Van Dijk, Marten, et al. "Fully homomorphic encryption over the integers." *Annual International Conference on the Theory and Applications of Cryptographic Techniques*.Springer, Berlin, Heidelberg, 2010.
21. Catteddu, Daniele. "Cloud Computing: benefits, risks and recommendations for information security." *Web application security*.Springer, Berlin, Heidelberg, 2010.17-17.
22. Gentry, C. "Fully homomorphic encryption using ideal lattices. Proceedings of the 41st annual ACM symposium on Symposium on theory of computing-STOC'09.Vol. 9." (2009).
23. Fontaine, Caroline, and Fabien Galand. "A survey of homomorphic encryption for nonspecialists." *EURASIP Journal on Information Security* 2007 (2007): 15.
24. ElGamal, Taher. "A public key cryptosystem and a signature scheme based on discrete logarithms." *IEEE transactions on information theory* 31.4 (1985): 469-472.
25. Yao, Andrew C. "Protocols for secure computations." *Foundations of Computer Science, 1982.SFCS'08.23rd Annual Symposium on.IEEE*, 1982.

