"Mircea cel Batran" Naval Academy Scientific Bulletin, Volume XX – 2017 – Issue 1
The journal is indexed in: PROQUEST / DOAJ / Crossref / EBSCOhost / INDEX COPERNICUS / DRJI / OAJI / JOURNAL INDEX / I2OR / SCIENCE LIBRARY INDEX / Google Scholar / Academic Keys/ ROAD Open Access / Academic Resources / Scientific Indexing Services / SCIPIO / JIFACTOR

# FUTURE DIRECTIONS IN CLOUD COMPUTING ENCRYPTION TECHNOLOGIES

**Sergiu EFTIMIE**[1]
**Vlad-Mihai COTENESCU**[2]
[1] Inf. Ph.D. Student Military Technical Academy - Electronic, Information and Communication Systems for Defense and Security Doctoral School, sergiu.eftimie@gmail.com
[2] Eng. Ph.D. Student Military Technical Academy - Electronic, Information and Communication Systems for Defense and Security Doctoral School, vlad.cotenescu@gmail.com

***Abstract:*** *From searchable to fully homomorphic encryption, this paper aims to provide an overview on the current and future developments in cloud encryption technologies. Advances in computation on encrypted data have led to new commercial services and there is an active ongoing research to further improve these new encryption techniques, while changing the industry.*
***Keywords:*** *Cloud, Encryption, Future developments*

## Introduction

Current developments in technology and the increasing focus towards mobility have changed the way we look at technology and its inherent issues. Businesses and individuals have begun to use external cloud servers managed by other companies in order to access data or computing services from any physical location. This move came with a major trust concern related to cloud providers.New ways of securing outsourced data have evolved along with the evolution of cloud computing.

The biggest risks that the consumers of cloud services are facing are related to the data disclosure or data loss[1].The benefits of cloud computing are significant: low cost, high reliability and immediate availability of additional computing resources when needed.Despite these advantages, both cloud service providers and consumers must be aware of their own set of unique risks of cloud computing, which is usually associated with storing and processing data.

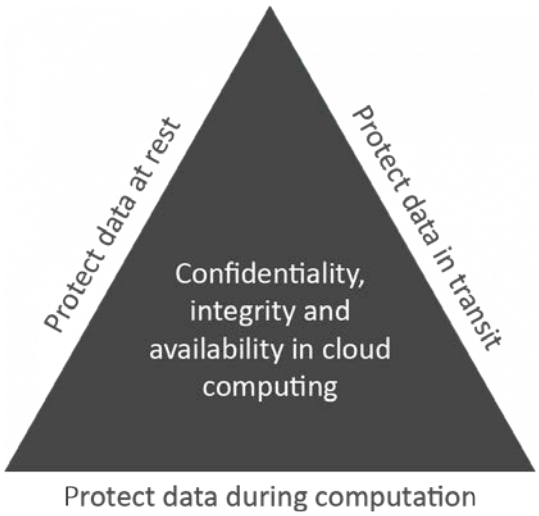


Fig. 1. The three pillars of cloud data protection

Users can encrypt their data before uploading it to an untrusted cloud service. This method works if the users intend to only use cloud services for storage, not for cloud services such as real-time data computation. For example, they may want to send a query to a search engine that would use an algorithm on the input data and return a result. In this case, data encryption makes it impossible to return a result. Consequently, to use the computing power of a cloud provider in a secure manner,the data encryption cannot be done using traditional techniques. In [Fig. 1.] we can observe the threepillars of data protection in the cloud.

The first twopillars, the protection of data at rest and the protection of data in transit have beenaddressed in various ways in traditional networks. The third pillar, the protection of data during computation has gained traction in the last decade along with some advances in cryptography.

## Hypothesis

In the context described above, we explore three forefront domains in cloud encryption, namely Searchable Encryption, Multi-Party Computation and Fully Homomorphic Encryption in order to document their current state and to establishdirections for future research.

## Searchable Encryption

In the recent years,there have been a series of incidents in which customer data hosted in the cloud was released online (for purposes of hacktivism and vandalism) or stolen for criminal purposes. Cloud computing is made possible through the use of technologies such as Internet access, virtualization and third-party data centers. In the case of online access to a cloud service provider, access controls take the form of usernames and passwords. In the case of virtualization, such access controls can be implemented through the logical separation of data. In the case of third-party datacenters, such access controls may take the form of physical access controls or software-based access to



DOI: 10.21279/1454-864X-17-I1-065

prevent unauthorized access to customer data of people working in the organization.

In principle, the access controls mentioned above are solid. However, in practice, such controls have been circumvented. If any of those access controls are compromised, the risk of data leakage is high. However, in the event of a data breach where the associated data is acquired in encrypted form, itis essentially useless for an attacker (unless the encryption algorithm used is weak and / or the attacker knows theassociated decryption key).

Otherwise, if a security breach occurs and the associated data in plain textisstolen by the attackers, the effects can be disastrous for a company, from negative publicity and damaged reputations to fines in accordance with the data protection legislation.

To reduce the impact of potential data breaches, cloud providers use cryptography. In a cloud environment, cryptography is commonly used for two purposes: security of data at rest and security of data in transit. Currently cloud cannot guarantee the security of data during processing due tothe current practical limitations of cryptography that prevent the processing of data in encrypted form. Given that the data is processed in its unencrypted form, a common approach is to targetdata in use rather than the data being encrypted at rest or in transit. An entity that wants to store its data in the cloud must choose whether to store datain encrypted or unencrypted form. In the first case in which it stores data in encrypted form, it can choose whether to disclose the decryption key to the cloud service provider or to keep the decryption key private. The disclosure of the decryption key must be donebecause the data cannotbe searched and processed in encrypted form. To provide this functionality to customers, cloud providers require access to the decryption key.

The second option is more secure. However, as previously mentioned, users lose the ability to browse or perform processing on their encrypted data. In order to use such a functionality in this case, users must download the data, decrypt it and then carry out those operations. Although this approach may be appropriate for small volumes of data, it is becoming increasingly ineffective as the amount of data processed increases. In addition, any changes to the data should be encrypted and uploaded back into the cloud. Clearly, none of the options listed provide an appropriate balance between data security and functionality. The first option offers full functionality but lacks security, while the second provides data security at the expense of functionality. The ideal solution for achieving an optimal balance of data security and functionality in the cloud services provider implies a capacity to search and manipulate encrypted data without holding the associated decryption key.

Searchable Symmetric Encryption (SSE) is one of the few forms of searchable encryption that can be implemented using standardized encryption algorithms[3]. Alternative forms of searchableencryption require the use of specially designed algorithms for this purpose. SSE is considered one of the less secure forms of searchable encryption, due to data leakage. There are solutions for this issue but they have a significant effect on the efficiency of searching for SSE. The challenge for researchers is to improve the SSE scheme securitywhile maintaining a high search efficiency[4].

Searchable encryption operates on the assumption that a term - either in plain-text or in encrypted form is located in the same position bothin the plain-text version of the document and the encrypted version. In essence, this description assumes that symmetric ciphers encrypt data one character at a time. In practice, this is not feasible for symmetric ciphers since modern onesencrypt data in fixed size blocks. The effect of using such ciphers is that the ciphertext associated with a term in plain-text is spread throughout the block of ciphertext.

Because of the inherent difficulty in achieving searchable encryption, the research work in this area focused on developing solutions for the original problem. Specifically, researchers focused on adapting an existing mechanism (reversed index) which was used in plain-text search algorithms. In its basic form, an index is a data structure that maps specific terms to the document(s) in which they appear, thus eliminating the need for sequential search. Adapting this technique to encrypted documents led to the creation of searchable symmetric encryption.

Searchable encryptionschemes allow cloud providers to process the search of strings provided by the user in encrypted documents without having to obtain information on search terms or the contents of the documents.

Currently, there are several different encryption schemes defined in the literature and their number is growing.

One of the main differences between the different searchable encryption schemes is the use of symmetric or asymmetric cryptographic techniques.

The symmetric approach is generally faster although the key management can become complex. Only the data owner has the ability to add documents to the data store. In the asymmetrical approach, anyone in possession of the owner's public key can add documents and



DOI: 10.21279/1454-864X-17-I1-065

only the owner can retrieve and search terms in the documents.
Another criterion for differentiating between searchable encryption schemes can be the search implementation. This search can be based on boolean values, complex phrases, regular expressions etc.
The search can be extended to the entire document or may be based on a small subset of keywords (based on indexes). When using index-based systems we can differentiate indexes per-document, where an index is assigned to a single document and system-levelindexes, where a single index contains all the words along with the links to the relevant documents. The system level implementation of the index makes the search generally faster, but updating the unique index may result in a significant consumption of resources, especially in the case of small changes made to documents.

**Fully Homomorphic Encryption**

The fully homomorphic encryption(FHE)is an encryption method that allowscomputational operations on ciphertext thereby generating an encrypted result which, when decrypted, matches the result of the calculations made on the original text. Craig Gentry introduced in September 2009 in his dissertation, a first fully homomorphic encryption scheme[2]. An intuitive example of homomorphic encryption [Fig. 2.] is the operation of concatenatingtwo texts. Using a substitution cipher, concatenation is possible using digital texts. Accordingly, the order in which concatenation and encryption operations are performed will be irrelevant and both approaches will lead to the same result.

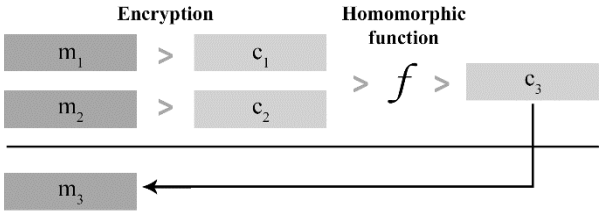


Fig. 2. Homomorphic Encryption

The fully homomorphic encryption is a cryptosystem which allows arithmetic operations (addition and multiplication) to be performed on encrypted data.
There are a number of partial homomorphic encryption schemes. RSA, one of the first public-key cryptographic algorithms,allows the computationof thesum of two plain-texts without decrypting them, although this is considered a vulnerability and it is eliminated by using padding functions. The ElGamal algorithm allows the product computation of two messages without decrypting them. The first major advance in this area was the discovery of the "somewhat" homomorphic schemes. They allow arbitrary operations on encrypted data but they introduce noise, and after a series of operations, theciphertext becomes indecipherable. Somewhat homomorphicschemesare usefulfor simple operations, but the revolutionary scheme of C. Gentry opened the possibility of a fully homomorphic encryption. His idea was to seek an encryption scheme with a low decryption complexity - a decryption operation that requires only a few operations. The operation of the decryption itself may be performed using somewhat homomorphic schemes. Such encryption schemes are called bootstrappable. Decryption eliminates noise and by decrypting the encrypted full text, we can carry out a "refresh" operation by generating a new ciphertext.
However, the goal is to obtain a refreshed cipher text method that does not require a secret key. Bootstrappingallows a noiseless ciphertext decryption on which we can perform further operations.
The first step in FHE is to build a bootstrappable somewhat homomorphic system. Gentry chose the lattice-based encryption, that is easy to deploy and decrypt using a low-complexity circuit. Most of today homomorphic schemes are based on lattices, although some alternatives are available. In this case, the decryption procedure involves only simple arithmetic operations.
The second step is to perform the refresh operation, namely the bootstrapping. In essence, the decryption operation is performed homomorphically using an encrypted secret key. Decryption will effectively remove the noise. This technique reintroduces noise but significantly less than was removed.
The disclosure of the encrypted secret key does not compromise security - bootstrapping is essentially a public operation. This double encryption is the starting point from which we can phase out the noise of encrypted texts.
Unfortunately the FHE method described above is highly inefficient and this has prevented its adoption. Bootstrapping operations are extremely costly in terms of computational power and must be performed fairly often and often digital texts are of considerable size. A new approach to build a FHE scheme is needed for practical implementations.

**Multi-party computation**

Multi-party computation is a form of computation that can be performed on encrypted data. In an MPC scheme a number of participants want to compute the value of a public function on their private data while keeping their inputs secret. The



DOI: 10.21279/1454-864X-17-I1-065

goal of MPC is to build an algorithm where the participants can obtain the result of the function without having to rely on a trustworthy third-party entity.MPC has also been established as the de facto paradigm for protecting privacy in distributed computation[6].

Although the technology to enable MPC has been present since 1980, only in the recent yearsprogresses have made it feasible for practical use.

Different MPC techniques exist, some based on Garbled Circuits and some based on secret sharing techniques.

Modern MPC protocols provide protection against the collusion attack where multiple participants develop a strategy to circumvent and break the protocol.

Due to the possibility of a collusion attack,two types of MPC can be developed: for honest majority and dishonest majority with different techniquesemployed for each one.

Generally speaking, the MPC tries to ensure both the input privacy and the correctness of the output. No information about the input data should be disclosed during the execution of the algorithm and any colluding parties should not be able to force an incorrect result.

Some examples of practical applications for MPC could be electronic auctions, e-voting[5], financial clearing, data mining etc.

**Conclusions**

Cloud computing has changed the business landscape and the current battle is fought in the realm of the information security.

Technologies such as fully homomorphic encryption, searchable encryption and multi-party computation have evolved since their initial feasibility studies, where in some cases the current advances in computing technologies improved their efficiency by several orders of magnitude. Despite these advances there is more work to be done in order for these technologies to reach practical use.

Fully Homomorphic Encryption is currently impractical. There is a need for fundamental research in this field. Multi-party computation is an interesting technology and could be a solution for e-government applications such as anonymous electronic voting. Searchable encryption is a promising research field with many practical applications and for that matter it is considered a near-to-market technology.

**Bibliography**

[1] Eftimie, Sergiu; Răcuciu, Ciprian: *Security Threats and Risks in Cloud Computing*, Sea-Conf 2015, Constanța, 2015.

[2] Gentry, Craig: *A fully homomorphic encryption scheme*, Stanford University, 2009

[3] Koschuch, Manuel; Hombauer Michael; Schefer-Wenzl, Sigrid; Habock, Ulrich; Hrdlicka, Stefan: *Fogging the Cloud - Implementing and Evaluating Searchable Encryption Schemes in Practice*, IFIP/IEEE International Symposium on Integrated Network Management, 2015

[4] Mc Brearty, Shaun; Farrelly,William; Curran, Kevin:*Preserving Data Privacy with Searchable Symmetric Encryption,* IEEE, 2016

[5] Naidu,P. Sanyasi; Kharat, Reena; Tekade, Ruchita,Mendhe, Pallavi;Magade, Varsha: *E-voting system using visual cryptography & secure multi-party computation,*IEEE Xplore, 2017

[6]Wang, Zhaohong; Cheung,Sen-Ching S.; Luo, Ying: *Information-Theoretic Secure Multi-Party Computation With Collusion Deterrence,* IEEE Transactions on Information Forensics and Security, 2017



**DOI: 10.21279/1454-864X-17-I1-065**