

# Data Confidentiality and Integrity in Cloud Storage Environment

Essohanam DJEKI<sup>1</sup>, Carlyna BONDIOMBOUY<sup>2</sup>, and Jules DEGILA<sup>3</sup>

<sup>1</sup> Institut de Mathmatiques et de Sciences Physiques, Porto-Novo, Benin  
essohanam.djeki@imsp-uac.org

<sup>2</sup> Centre d'Excellence Africain - Science Mathmatiques et Applications,  
Benin carlynablessing@gmail.com

<sup>3</sup> Institut de Mathmatiques et de Sciences Physiques, Porto-Novo, Benin  
jules.degila@imsp-uac.org

**Abstract.** Cloud services have seen a considerable increase in recent times, as the cloud allows users to outsource their data and IT resources. Storing and backing up data in the cloud has become increasingly popular. However, data security is an increasing concern. With the number of attacks on the cloud and data leakage, users are worried about their data security in the cloud. Several works dealt with data security in the cloud, but most of these solutions largely depend on providers. They do not provide users the control of the security of their data. To deal with data security concerns, we proposed a solution called EnCrypt Cloud that allows users to encryptcheck the integrity of their files, check their files' integrity, and check their files' integrity before uploading and storing it the cloud storage. We used the encryption technique (AES 256) to ensure the confidentiality of the data. To verify the data's integrity, we used the SHA 256 hash function with a two-level integrity check. Performance analysis of the AES encryption algorithm was performed to compare execution time memory usage during the encryption and decryption process. It should be noted that decryption consumes more resources than encryption.

**Keywords:** data confidentiality, data integrity, cloud storage, data encryption, cloud security issues, EnCrypt Cloud.

## 1 Introduction

Cloud computing is a concept that represents access information and services located on a remote server via Internet. Cloud computing has the potential to revolutionize IT, but it has some disadvantages, like security threats. With the expansion of the cloud, development of its various services and technologies used, new security issues have arisen. Indeed, safety has been a hot topic during forums on privacy and data governance. The controversial issue of confidentiality and data security remains the cloud's major limitation [1]. The risk of seeing the data in a situation of theft, illegal use, or unauthorized use is possible. From an information security perspective, questions are raised about the various threats

facing the cloud, especially: what are the risks of outsourcing data in the cloud, and how can cloud customers ensure their data security?

In this paper, we were interested in data confidentiality and integrity in the cloud. To address this problem, several research efforts have been undertaken in recent years to secure and protect data in the cloud and secure and protect data in the cloud and secure and protect data in the cloud and secure and protect data in the cloud. Still, complete security is far from being obvious because of the diversity of possible problems and attacks. We will first start with a cloud overview and its security issues. We will then look at the related works to see what is already achieved for data security in the cloud storage. We will end with our proposed solution to solve problems related to data protection in transit or at repository, and its performance analysis.

## 2 Cloud Computing

### 2.1 Overview

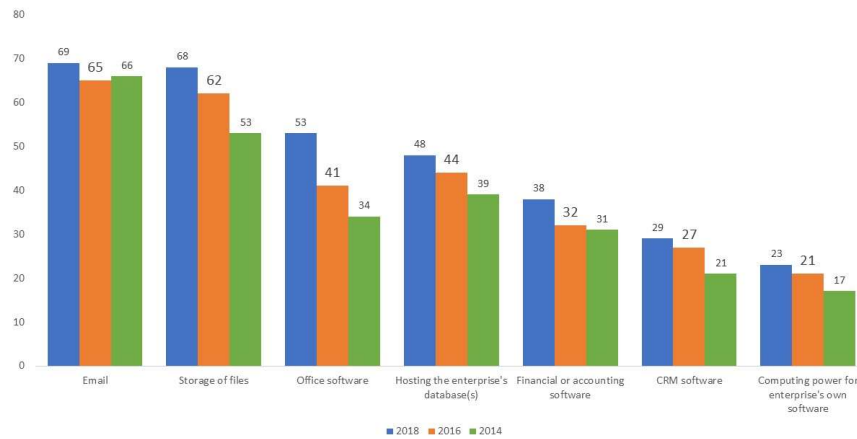
In 2018, 26% of European Economic Community enterprises used cloud computing, mostly for hosting their email systems (69%), storing files in electronic form (68%), office software (53%), databases (48%), financial or accounting software applications (38%), CRM software (29%), and computing power for "enterprise's software (23%) [2]. Fig. 1 shows the comparison between 2014, 2016, and 2018 in using cloud services by purpose. The use of cloud in 2018 for email and file storage is still predominant. The use of office software has recorded the highest growth (+19%) since 2014, among all purposes. The more sophisticated purposes of cloud services (for financial and accounting software applications, CRM software applications, and Computing power) recorded smaller increases (+7%, +8%, and +6%, respectively) [3].

### 2.2 Security Issues

The cloud has advantages, but it also has some disadvantages, including security issues, i.e., data security issues, networking issues, virtualization issues, organization security management, and backup and recovery issues. In the case of data security, we identified some "cloud's challenges such as data confidentiality, data integrity, data availability, data storage, data lock-in, data breaches, data access, data locality, data privacy, etc. According to Eurostat [2], companies' highest risk is the security breaches, e.g., 57% for large companies and 38% for SME (see Fig. 2).

We all know that even a 1% security breaches are enough for a criminal hacker to initiate an attack. This is why organizations are increasingly placing more importance on the security of their IT systems. But protection is much more

difficult in the cloud environment because of the outsourcing of the system, or servers, or applications leading to loss of control over the system, and the data. Other issues faced by organizations using the cloud are data governance (46% for large enterprises and 31% for SME), data location (46% for large enterprises and 29% for SME), and problems accessing data or



**Fig.1.** Cloud computing adoption in services

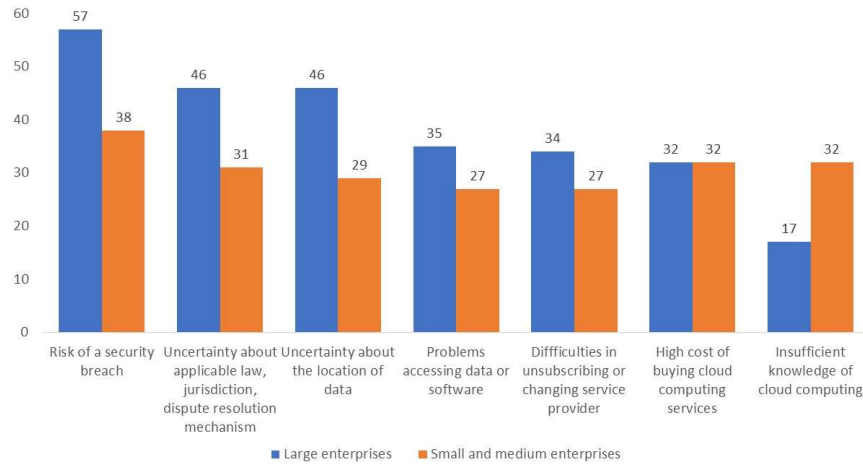
services (35% for large enterprises, 27% for SME), insufficient knowledge, or ICT [2]. Thirty-two percent of SMEs using the cloud reported that the problems listed above were a factor limiting the cloud's adoption, compared with 17% of large enterprises. An observation of Fig. 2 shows that security breaches are on the rise.

Data breaches in the United States reached a six-month high of 29% in June 2017, up from the same period the previous year, according to the Identity Theft Resource Centre Report (ITRC) [4]. The Equifax data breach example caused considerable damage by exposing "users' sensitive personal data, such as names, dates of birth, social security numbers, " driver's license numbers, credit card numbers, etc., to the public. 'IBM's study showed that data breaches come from 53% of insiders and inadvertent of 'CSP's employees, 42% by outsiders (criminal hackers), and only 5% are caused by insiders malicious [4].

The cloud has excellent potential due to its advantages and has its challenges, among which are security problems. Given a large number of data leaks and the origin of the causes, it is clear that nothing can be done to prevent data leakage 100%. That is why more efforts must be made to enable cloud users to secure their data in transit and at storage. In the next section, we will explore related work in this direction.

### 3 Related Works

Data protection often refers to laws concerning the regulation of personal data and privacy. Many states already established data security regulations to prevent abuse and protect users from potential risks. F. Pfarr, T. Buckel, and A.



**Fig.2.** Cloud Computing popular limits

Winkelmann [5] highlighted the different necessary current rules to protect the data and privacy of Internet users' data and privacy that can be applied in cloud environments. Among these regulations, the authors talked about the proposed US Safe Harbor Agreement, the EU General Data Protection Regulation (GDPR), and also the Binding Corporate Rules, which is a regulation ensuring the security of data exchanged between EU countries and third world countries. Laws vary from one country to another depending on the government. While the Internet has no borders and the duplication of data in geographic areas used by the cloud to ensure data availability, it is becoming increasingly difficult to enforce a law on replicated data.

Ni et al. [6] worked on a new model to ensure data integrity and cloud storage confidentiality. The authors propose a new secure outsourced data transfer scheme (SODT) based on polynomial-based authenticators and the BCP encryption scheme. They enhance the BCP encryption scheme to handle proxy re-encryption and use it to maintain the outsourced data's confidentiality. Using the homomorphism of the enhanced BCP encryption scheme and polynomial-based authenticators, users can efficiently verify the integrity of their data in the cloud to ensure the security of outsourced data transfer in the cloud. Users can also

discard data transferred to the cloud by proxy re-encryption method. Their solution focuses on the transfer or migration of data from a cloud provider to a new provider, so the solution is used to ensure data integrity during the migration and ensure that these data have been adequately removed from the old provider to maintain data confidentiality.

S. Han and J. Xing [7] design a novel third party auditor scheme in cloud storage. Their solution is composed of two parts, Users and Advanced Cloud Service Provider (ACSP) in which the third-party auditor function combines with the Cloud Service Provider. To ensure data security, the authors use the RSA algorithm to encrypt all data flow between servers in the ACSP and Bilinear Diffie Hellman to ensure security while exchanging keys between users and servers in their scheme. A limitation to the cryptographic solution is that these CSPs know the encryption and decryption keys to access their customers' data. The cryptographic techniques add computational overhead to CSP, while their primary objective is optimal resource use.

To solve this problem, I. Zakaria, H. Mustapha, and B. Igarramen [8] have thought of using File Assured Deletion (FADE), which is a promising solution to protect user data. FADE ensures the deletion of files by making them unrecoverable for anyone, including cloud storage providers. The system is built by encrypting all data files before outsourcing, then using a trusted party to outsource the key cryptography. But this methodology remains weak because its security largely depends on the key manager's security. To overcome this problem, the authors have proposed a new system to improve the security of the FADE using the Trusted Platform Module (TPM), and the Encrypted File System (EFS).

To protect data from unauthorized users access, the data are supposed to be either encrypted or unreadable. R. Sugumar, and K. Arul Marie Joyce [9] propose an obfuscation technique for enhancing cloud storage data. This obfuscation technique is used to encrypt and decrypt all data. The authors propose a confidentiality system called SUG-DO (SUGUMAR Digits Obfuscation) to strengthen data security in the cloud environment. The obfuscation technique cannot be used to secure any data or large data.

G. S. Mahmood, D. J. Huang, and B. A. Jaleel [10] propose a new cloud storage system to enhance data security (images), while ensuring a secure and reliable cloud storage service. A secret image is encrypted using the AES algorithm. The encrypted image embedded into the host image using a steganography technique, which combines Discrete Wavelet Transformation (DWT), and Singular Value Decomposition (SVD) to produce the stego image. To preserve the data's integrity, a hash value is computed for the stego image using the SHA-512 before storing the image in the cloud. Once the image is retrieved from the cloud, its hash value is generated using the same algorithm (i.e., SHA-512). The two hash values are then compared to check if the data stored in the cloud changes and to get the secret image.

To ensure the confidentiality of outsourced data, S. S. Manikandasaran, L. Arockiam, and P.D. Sheba Kezia Malarchelvi [11] propose a technique called

MONcrypt, which is based on obfuscation technique. Generally, obfuscation is done without using any key to obfuscate "users' data, but the authors have developed a method that uses a key for de-obfuscation. During the process of obfuscation, a count value is produced from the original text. This count value is used as a de-obfuscation key. The key is stored as metadata in the client part. The authors conducted an experience with sample data, and performance analysed in terms of time taken for obfuscation and de-obfuscation shows that the MONcrypt offers maximum security for outsourced data in less time. Unfortunately, MONcrypt is only applied to numerical data.

M. S. Abolghasemi, M. M. Sefidab, and R. E. Atani [12] worked on improving the data encryption technique used in cloud storage. The authors proposed a data encryption method based on the " user's geolocation, called Geo Encryption, to add a new security layer to the existing security measures. AntiSpoof, and accurate GPS is required to give the latitude and longitude accurately. The Geo Encryption algorithm is implemented in the cloud, and the " user's computer, which is connected to the GPS, is used to encrypt - decrypt the data. When uploading data to the cloud, the spatial location is used to generate the encryption and decryption key. To retrieve and decrypt the data, the user had to be in the same location. As a result, they can limit the data access to a specific location at a particular time. Their solution is more appropriate for banks, big companies, institutions, and examples like this.

S. K. Sood [13] proposed a model that combined various techniques to perform data security in the cloud. His solution uses encryption as the main fundamental protection scheme. The framework introduces the notion of classification of data based on Sensitivity Rate (SR), which is computed from the combination of confidentiality, availability, and integrity provided by the user himself, and according to SR, the system decide where to upload the data (Public, Private, or Owner limited access storage), e.g., if  $SR < 3 \Rightarrow$  Public,  $3 < SR \leq 6 \Rightarrow$  Private, and  $6 < SR \leq 10 \Rightarrow$  Owner limited access storage. His solution provides data availability by surpassing many issues like data leakage, tampering of data, and unauthorized access even from the CSP, as he says.

R. Pitchai et al. [14] proposed a new protocol called Availability and Integrity Verification Protocol (AIVP) to predict the available space in the cloud and verify the integrity of the data stored in it. The authors have separated public and private data for integrity checking, thus ensuring data confidentiality. The -service providers in the cloud. They upload the private data, and the public data will be uploaded by the third-party auditor. In the results of the simulation, the AIVP proposed by the authors outperformed the higher performance and throughput, which they say will reduce latency, computing costs, and communication costs.

P. Wei et al. [15] conducted a study on the use of blockchain techniques to improve data integrity checking by reducing the excessive complexity of calculations or lack of scalability. The authors deployed a model of a distributed virtual machine agent in the cloud using mobile agent technology. The virtual machine agent allows multiple tenants to cooperate to ensure data trust verification. The integrity

protection framework based on the blockchain is constructed by the proxy model of the virtual machine. The unique hash value corresponding to the file generated by the Merkel hash tree is used to monitor data change through the intelligent contract on the blocking chain, and data is held on time; Besides, a “block-and-response” mode is used to build a data integrity verification scheme in the cloud based on the blocking chain.

A. Li, S. Tan, and Y. Jia [16] proposed a new method for proving data integrity (PDI) for customers whose data is hosted on non-trusted servers in cloud computing. An advantage of their model is low cost to the customer since a consistent volume of metadata is generated. The authors propose a simple and efficient audit service for public verification of non-trusted outsourced storage based on a bilinear group. In comparison to existing PDI methods, they aim to ensure integrity by considering the cost of generating the audit metadata at the customer’s side. Besides, their method supports data dynamics and public verifiability.

## 4 Our Proposed Model

With the growing number of collaboration tools which require resources (files) to be shared, artificial intelligence that involves data (structured and unstructured) for learning, big data, and also the emergence of the Bring Your Device concept, which requires resources (data, or files) to be accessible at any time, from anywhere. We estimate that the use of the cloud for storage is on any device and will be increasing. For these reasons, we have decided to make our contribution to data (files) protection in the cloud and mostly focused on the client-side.

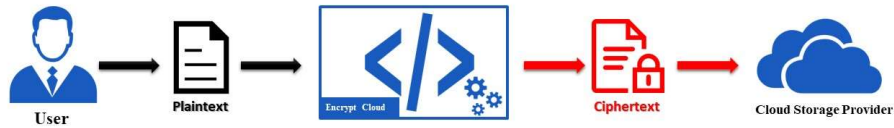
With the traditional architecture, when a user sends files to the cloud, the file is transmitted and stored in plain text. An unauthorized person can intercept it during the sending (man in the middle attacks) or even access the files in the pooling cloud. Sometimes the connection with the “CSPs’ servers is made securely, i.e., encrypted, they use security protocols such as SSL/TLS, SSH, IPsec or VPN, etc., but the data sent is stored in plain text on the “CSPs’ storage servers. According to the Service Level Agreement (SLA) of some providers, the data is encrypted in storage, there is still a problem with the encryption-decryption keys. This stipulates that the storage provider has access to the encryption-decryption keys; with this information, the provider can illegally access and use “users’ data without their permission and consent. Our solution aims to enable users to encrypt their data (files) before uploading it. Users are, therefore, in control of the security of their data. It is an application solution we called Encrypt cloud, i.e., it is about offering an application that allows users to encrypt and verify the integrity files before sending them to the cloud.

At the macro level, Fig. 3. shows the architecture. In the proposed architecture, users interact with an interface that acts as an intermediary between the users and their CSP by integrating an additional security layer. Before uploading a file, the

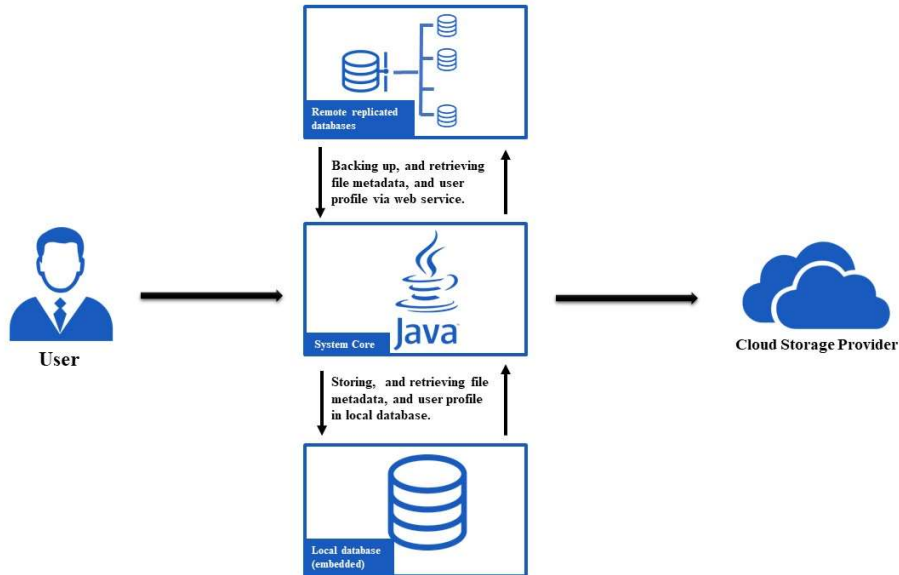
user chooses whether he wants to encrypt or not or whether he wants to check the file's integrity or whether he wants to check the file's integrity.

The Encrypt Cloud architecture is shown in Fig. 4. Encrypt cloud consists of three components: *System Core*, *Local Database*, and *Remote Replicated Databases*.

The *System Core* is the engine of the application, it uses both components to



**Fig.3.** Generic architecture of the proposition



**Fig.4.** The architecture of Encrypt Cloud

ensure effective processing of the application.

The *Local Database* is the data repository and used to store all high-level data; the aim is to allow users to work offline. The storage system is an embedded database, and all tables follow a certain degree of database standardization. There are two



main entities in the database, namely, *user profile* and *file metadata*. The *user profile* contains information about the system's users, including user id, username, password, e-mail, csp, access token, which allows linking their account to the cloud, among other personal information. *File metadata* store metadata about each file uploaded by the user. This information includes user id, file name, the date and time, hash of the file, encryption key associated with the file, and other information. Why associate an encryption key with each file? We could have generated a single encryption key for users. Still, we would like to share files between users without compromising the owners' securityshared files' owners. By assigning an encryption key to each file, a file can be shared without worrying about other files' protection.

The *Remote Replicated Databases* provides the availability and continuity of the service and data also. Before starting, users must create an account by providing a username, password, recovery e-mail, and information about their CSP. This information is stored locally (an embedded database). Suppose the user loses his machine, or reinstalls it, or wants to access data using another machine or device, he or she will no longer be able to use his encrypted data or check the integrity of his data, because he no longer has the encryption-decryption key, or the hashes of his files. The Remote Replicated Databases will therefore be used to save user information and file metadata remotely.

When a user sends a file, depending on his choice, the system will decide to calculate and save the 's hash using SHA-256 the file's integrity. With " user's information, the file will be encrypted using the AES-256 (Advanced Encryption Standard) algorithm according to his choice. The encrypted file will be sent to the " user's appropriate cloud using the API published by his CSP, and the metadata will be saved on local, and remote database as we can see on Algorithm 1.

---

**Algorithm 1** File uploading process

---

- 1: Select the file to be uploaded from local storage.
  - 2: **if the** user chooses to check the file's integrity, **then** 3: Compute hash(file).
  - 4: **if the** user chooses to encrypt the file, **then** 5:  
Generate an encryption key, and encrypt(file).
  - 6: Send the file to the cloud using 'CSP's API.
  - 7: Save 'file's metadata in the local database and remotely replicated databases.
- 

After authenticating, the user can retrieve, exploit, or download his files. To download a file, as described with Algorithm 2, the user selects the file he wants; the system recovers the file metadata; if the file is encrypted, it is decrypted with AES-256. Encrypt cloud uses a double integrity check to ensure the files' integrityfiles' integrity and the databases containing the metadata. Once we have

the file's metadata, we compare the signatures contained in the databases, if a database is corrupted, i.e., all signatures " don't match, we then make a hash comparison and choose the majority hash. Finally, we calculate the hash of the file and compare it with the hash from the databases.

---

**Algorithm 2** File downloading process

---

- 1: Select the file to be downloaded.
  - 2: Retrieve 'file's metadata (including the signature and encrypt - decryption key) from the local database and from remote replicated databases.
  - 3: **if** the file is encrypted, **then**
  - 4: decrypt(file).
  - 5: Compare local signature (from local database) with remote signatures (from replicated databases). If all signatures " don't match, then select the majority signature.
  - 6: Compute hash or signature of the downloaded file.
  - 7: Compare the majority hash with the downloaded 'file's hash.
- 

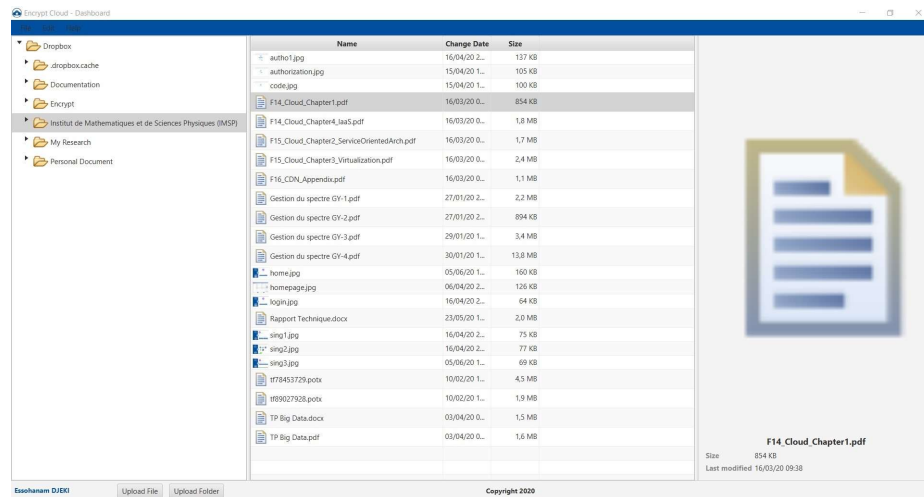
## 5 Sample User Sessions Using Encrypt Cloud

In this section, we provide detailed examples to illustrate how a user interacts with *Encrypt Cloud*. You must log in by providing a username and password before using the application. You must have an account. The creation process follows three steps:

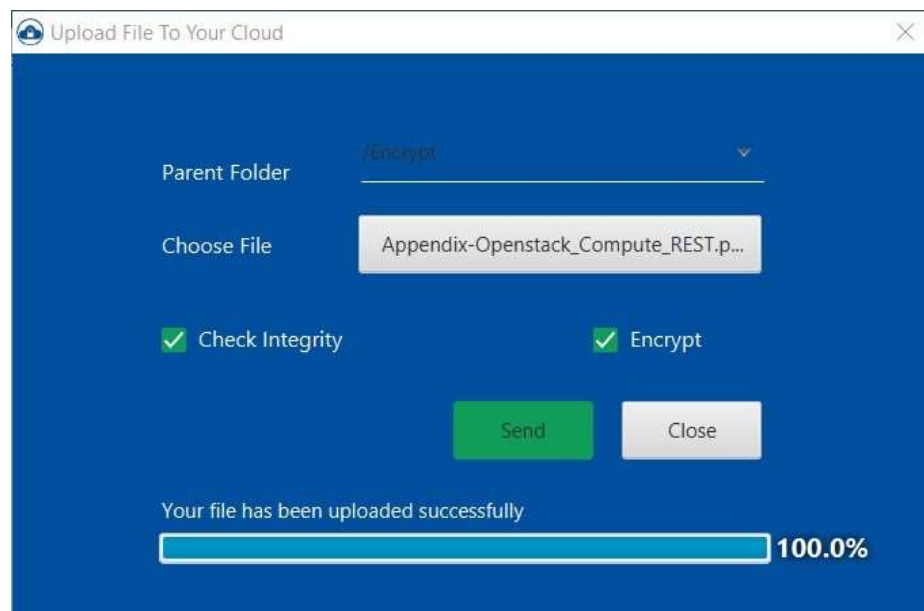
- First step: provide your profile information, i.e., username, recovery e-mail, and password;
- Second step: select your cloud provider;
- Third step: link your cloud account to your *Encrypt Cloud* account. Once you have selected your provider, a web page will ask you to log in to allow your cloud account to be linked to *Encrypt Cloud*. Your provider will provide you with an authorization code that you will copy and paste into the *Encrypt Cloud*, and click on "Finish" "button to finish the process.

Once the account is successfully created, the application automatically takes you back to the authentication page. Log in with your credentials, and you will be on the home page (Fig. 5).

The home page is mainly composed of three columns: the first column presents the " user's folders; the second column presents the files and/or sub-folders of the parent directory, and the last one presents a detailed view of the items. From the home page, we can send files via the "" Upload File" "button or send a whole folder via the ""Upload Folder" "button.



**Fig.5.** Encrypt Cloud Home Page



**Fig.6.** File Uploading Window

Once you click on the “Upload File” button, a window will open (Fig. 6.), you will have to specify the folder where the file(s) will be stored, choose the file(s) from your machine, decide whether to encrypt the file or not and decide if the integrity

of the file(s) should be checked. In our case, we sent the file " *AppendixOpenstack Compute REST.pdf* " in Dropbox, we encrypted it and verified its integrity. A double click on a file opens a window displaying the file's information, whether the file is encrypted or not, whether the file integrity check is enabled or not, whether the file has been altered or not. Fig. 7 shows that the file has been altered. The system has detected that the file has been modified. The system asks to know if we are at the origin of the modification, if yes, we have the possibility to update the hash by checking " "Yes" " or " "No" " otherwise.



Fig.7. File information Window

## 6 Other Encryption Tools

We have found some applications that offer solutions acceptable. Among these solutions, we have: EnCrypted Cloud, which shifts your Google Drive, Dropbox, Box, and Egnyte folders into an enCrypted folder; Encrypto, which focuses on sending, and sharing (via USB, or e-mail) encrypted files, but also works for encryption of local, and cloud storage (Dropbox); Boxcryptor, you can create a folder anywhere on your computer, and everything in that folder is encrypted, and in case you use a cloud storage service like Google Drive, or Dropbox, this folder

can be placed in the synchronization folder of the corresponding application; Sookasa works very similarly to BoxCryptor; Cryptomator saves the files in an encrypted folder in the vault somewhere on your computer, or on your Google Drive, or Dropbox folder; NordLocker works similarly to Cryptomator; Odrive is a desktop, and web tool which enables you to link all your online storage accounts (Google Drive, or Dropbox), its particularity is that files are only encrypted on the cloud storage, not on the local computer; AxCrypt; BitLocker; Perfecto Encryptor; VeraCrypt, and EncFSMP.

These encryption tools use the AES encryption algorithm (Table 2), are available on almost all platforms, generally on Windows, Mac, Android, iOS, and sometimes on Linux, and available in free or evaluation version and paid version (see Table 1).

We have noticed that most of these tools are applications that are specifically designed to manage files by introducing the notion of “Vault.” And that these vaults can be synchronized with cloud accounts. These vaults are protected by a unique password, which means that the vault remains inaccessible in the event of loss or forgotten password files are lost. Beyond its limitations, we have noticed that these tools do not support file integrity checking (see Table 2).

**Table 1.** Encryption Tools: Platforms, and Prices.

Software	Platforms	Pricing
EnCrypted Cloud	Win, Mac, Android, and iOS.	Free
Encrypto	Windows, and Mac.	Free
Boxcryptor	Win, Mac, Android, and iOS.	Free
Sookasa	Win, Mac, Android, and iOS.	Free
Cryptomator	Win, Mac, Android, and iOS.	Premium (\$10 /month)
NordLocker	Win, Mac.	Free (Pay what you want)
Odrive	Win, Mac, Linux.	Free (5Go) Premium (\$2.99/month)
VeraCrypt	Win, Mac, Linux.	Free - Premium
Perfecto Encryptor	Win, Mac, Linux.	Free
EncFSMP	Win, Mac, Linux.	Free
AxCrypt	Win, Mac, Android, and iOS.	Free
<b>EnCrypt Cloud</b>	<b>Win, Mac, Linux.</b>	<b>Free*</b>

## 7 Performance Analysis

The performance of systems or applications needs to be evaluated based on a number of criteria. These criteria determine whether a system or application is

**Table 2.** Encryption Tools Features.

Software	Confidentiality	Integrity
EnCrypted Cloud	Yes (AES 256)	No

Encrypto	Yes (AES 256)	No
Boxcryptor	Yes (AES 256)	No
Sookasa	Yes (AES 256)	No
Cryptomator	Yes (AES 256)	No
NordLocker	Yes (AES 256 - RSA 4095)	No
Odrive	Yes (AES 256)	No
VeraCrypt	Yes (AES 256)	No
Perfecto Encryptor	Yes (AES 256)	No
EncFSMP	Yes (AES 256)	No
AxCrypt	Yes (AES 256)	No
<b>EnCrypt Cloud</b>	<b>Yes (AES 256)</b>	<b>Yes (SHA 256)</b>

efficient/effective or not: these parameters are known as performance metrics. To analyze our solution's performance applications performance our solution's performance. We need to analyze the performance of the application when uploading and downloading files. Uploading and downloading files depends on encryption and network speed. Since the network speed varies from one user to another, we have analyzed the performance of the AES algorithm encryption while sending and receiving based on the metrics used by A. Lemma, M. Tolentino, and G. Mehari [17], which are:

- Encryption and Decryption Time: the execution time it takes for an encryption algorithm to produce ciphertext from plain text or plain text from ciphertext;
- Encryption and Decryption Memory Usage: is the amount of memory (RAM) consumed when the encryption or decryption process;
- Encryption and Decryption Throughput: ~~is~~ are calculated from the file (plain text or ciphertext) size and the time it takes to processing encryption or decryption using the following formula:

$$Throughput = \frac{TextSize}{ExecutionTime}$$

Many works in the literature have been carried out to make a comparative study of the performance of encryption algorithms. Their work generally consists of making a comparative study between the encryption algorithms to know which one is more efficient and with which metric(s). Their studies have shown that AES and RSA are the most efficient, RSA consumes more resources but is more secure than others, including AES, and AES consumes fewer resources, which is why all encryption tools use AES (see Table 2). Our performance analysis focuses on AES by comparing the performance of encryption and decryption regarding execution time, memory used, and throughput.

The machine used for the simulation described in this paper is a Core i7 CPU @ 2.70 GHz, 8GB RAM, Processor x64, Windows 10. The solution we propose is developed in Java Fx, so the encryption and decryption performance analysis is simulated with JDK (jdk1.8.0 172) with NetBeans IDE 8.2. To evaluate the performance, we generated files with sizes ranging from 1024KB to 102400KB,

and each time a file is encrypted or decrypted. We collect the data (execution time and memory usage) that we save in an excel file that allowed us to draw charts.

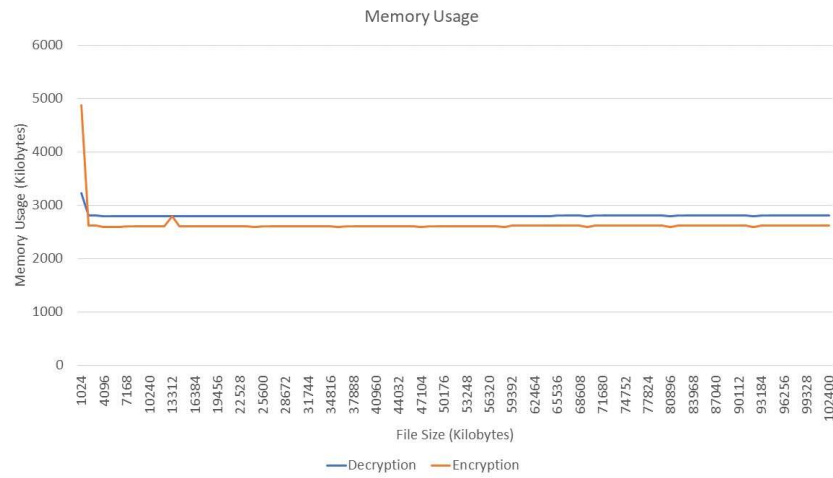
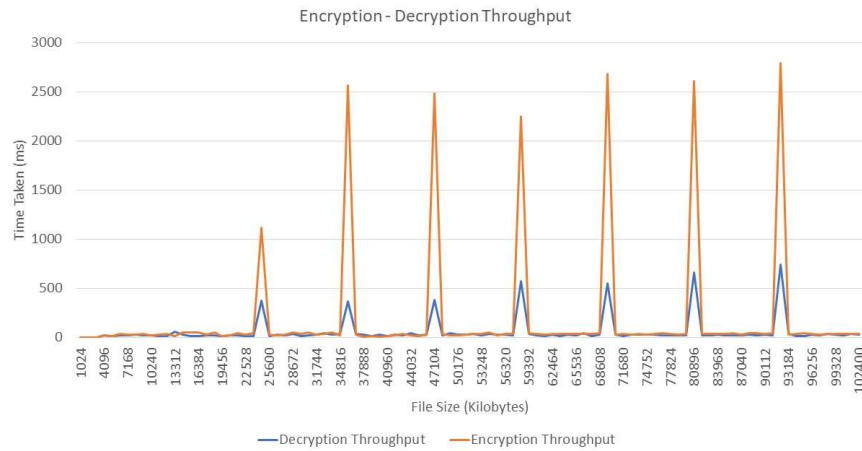


**Fig.8.** Execution Time

Fig. 8 shows the comparison between the encryption and decryption execution time relative to the files' size. We can see that the encryption and decryption times have the same trends, the times are almost the same for sizes between 1MB - 59MB, and that the decryption time is higher than the encryption time for files larger than 60MB. We also noticed that as the files' size increases, the encryption time increases slowly while the decryption time increases rapidly.

Fig. 9 shows the memory used for encryption and decryption; we can see that the size used is more or less stable and that the memory used for decryption is slightly higher than the memory used for encryption. A little surprising thing that we can observe on Fig. 8 and Fig. 9 is the high peaks at the beginning; for 1024KB, we notice an encryption time of 9817ms and 4876KB of memory usage, and 7435ms (execution time) and 3220Kb (memory usage) for decryption. You may be surprised by this result. But it makes sense because of the "javax.crypto.Cipher" "class needs a little "warm-up" "when it is first used in the program.

The throughput results of the encryption and decryption is shown in Fig. 10,

**Fig.9.** Memory Usage**Fig.10.** Throughput

these results show that the decrypting ~~decryption~~ process has the lowest throughput; while the encryption process has the highest throughput.



## 8 Conclusion

In this paper, we proposed an architecture primarily based on encryption to secure data during its transmission over the network and its storage ~~on~~ in the cloud. Thus, we have introduced a third party, which is thus introduced a third party. This application acts as an intermediary between the user and his cloud by providing an additional security layer. This application has been designed to be implemented at the "customers' level to give them the possibility to ensure the security of their files ~~on~~ in the cloud. Several tools are available for encrypting files before sending them to the cloud. However, they do not allow users to know whether their files are corrupted or not. Encrypt cloud, which we are proposing, brings a new functionality, offering users the possibility to check the integrity of their files by proposing a double-checking approach of file. Users can check the integrity of their files by proposing a double-checking approach of file. Users can check the integrity of their files by proposing a double-checking approach of file and database integrity. Another feature our solution provides is encrypted file sharing, allowing users to collaborate on files without compromising the security of other files. Users can also encrypt their sensitive data (files) to ensure that no unauthorized person can exploit it, even in a data leak. To encrypt the data, we have used AES 256, which is asymmetrical encryption algorithm because we will have to encrypt files of varying sizes ranging from MB to GB. To evaluate our system's performance, we performed a performance analysis based on execution time, memory usageour system's performance performed a performance analysis based on execution time, memory usage, and throughput of encryption - decryption using files ranging in size from 1024KB to 102400KB. This analysis showed that the execution time of encryption and decryption is almost the same for files smaller than 60MB. The execution time of decryption is higher than the execution time of encryption for files larger than 60MB. The performance study also showed that memory usage is more or less stable for both processes (between 2595KB and 2794KB), and that the encryption throughput is high compared to the decryption throughput. It should be noted that decryption consumes more resources than encryption in the case of AES. Currently, all information, especially encryption and decryption keys and signatures are stored in a database, whether local or replicated, is a single point of failure, making the solution vulnerable. In our future work, we will work on the architecture of the solution to decentralize the information.

## References

1. S. Kaur, and A. Kaur, ""Survey of security algorithms in cloud, 2015
2. Eurostat, [https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Archive:Cloud computing \\_statistics on the use by enterprises \\_2016 data](https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Archive:Cloud_computing_statistics_on_the_use_by_enterprises_2016_data)

3. K. Magdalena, and S. Maria, ""Cloud Computing - statistics on the use by enterprises, December 2018, [https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Cloud computing \\_statistics on the use by enterprises# Use of cloud computing: highlights](https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Cloud_computing_statistics_on_the_use_by_enterprises#_Use_of_cloud_computing_highlights)
4. UnfoldLabs, ""8 Trends in Cloud Computing for 2018, November 2017, <https://medium.com/@Unfoldlabs/8-trends-in-cloud-computing-for-2018-d893be2d8989>
5. F. Pfarr, T. Buckel, and A. Winkelmann, ""Cloud computing data protection literature review and analysis, in 2014 47th Hawaii International Conference on System Sciences. IEEE, 2014, pp. 50185027.
6. J. Ni, X. Lin, K. Zhang, Y. Yu, and X. S. Shen, ""Secure outsourced data transfer with integrity verification in cloud storage," in 2016 IEEE/CIC International Conference on Communications in China (ICCC). IEEE, 2016, pp. 1–6.
7. S. Han and J. Xing, ""Ensuring data storage security through a novel third party auditor scheme in cloud computing, in 2011 IEEE International Conference on Cloud Computing and Intelligence Systems. IEEE, 2011, pp. 264268.
8. I. Zakaria, H. Mustapha, and B. Igarramen, ""A DATA CONFIDENTIALITY SYSTEM BASED ON TRUSTED PLATFORM MODULE IN CLOUD STORAGE ENVIRONMENT, in 2019 Journal of Theoretical and Applied Information Technology.
9. R. Sugumar, and K. Arul Marie Joyce, ""Ensure and Secure Data Confidentiality in Cloud Computing Environment using Data Obfuscation Technique, in International Journal of Advanced Studies in Computers, Science and Engineering. 2017, pp. 16–21.
10. G. S. Mahmood, D. J. Huang, and B. A. Jaleel, ""Achieving an Effective, Confidentiality and Integrity of Data in Cloud Computing, in IJ Network Security. 2019, pp. 326–332.
11. S. S. Manikandasaran, L. Arockiam, and P.D. Sheba Kezia Malarchelvi, ""MONcrypt: a technique to ensure the confidentiality of outsourced data in cloud storage, in International Journal of Information and Computer Security. Inderscience Publishers (IEL), 2019, pp. 1–16.
12. M. S. Abolghasemi, M. M. Sefidab, and R. E. Atani, ""Using location-based encryption to improve the security of data access in cloud computing, in 2013 international conference on advances in computing, communications and informatics (ICACCI). IEEE, 2013, pp. 261265.
13. S. K. Sood, ""A combined approach to ensure data security in cloud computing, in Journal of Network and Computer Applications, vol. 35, no. 6, pp. 18311838, 2012.
14. Pitchai, R., Babu, S., Supraja, P. et al. " "Prediction of availability and integrity of cloud data using soft computing technique," Soft Comput 23, 85558562 (2019). <https://doi.org/10.1007/s00500-019-04008-0>.
15. Wei, PengCheng, et al. " "Blockchain data-based cloud data integrity protection mechanism," Future Generation Computer Systems 102, 2020, pp. 902–911.
16. Li, Aiping, Shuang Tan, and Yan Jia, ""A method for achieving provable data integrity in cloud computing," The Journal of Supercomputing 75.1, 2019, pp. 92–108.
17. A. Lemma, M. Tolentino, and G. Mehari, ""Performance Analysis on the Implementation of Data Encryption Algorithms Used in Network Security, International Journal of Computer and Information Technology, vol. 04, no. 04, pp. 711–717, 2015.