ENHANCING SECURITY IN THE CLOUD USING ENCRYPTION IN A CLIENT CENTRIC ACCESS CONTROL MECHANISM

Amaniampong Adjei[1], Francis Ofori[1], Beatrice Birago[1], and Isaac Ofori[2]

[1]Department of ICT and Mathematics, Offinso University College of Education, Post Office Box 7, Offinso Ghana.

[2]Department of Science, Offinso University College of Education, Post Office Box 7, Offinso Ghana.

ABSTRACT: *In this paper, a security solution for data storage in cloud computing is examined. The solution encompasses confidentiality and integrity of the stored data, as well as a secured data sharing mechanism in the cloud storage systems. For this purpose, cryptographic access control is used, which is mainly based on cryptography. Based on an analysis of the cryptographic access control mechanism, a design of encryption algorithm is created for a system, which is intended to demonstrate the security mechanism in practice. Finally, on the basis of the proposed design of encryption algorithm which is created, a prototype is implemented. Data confidentiality and integrity are ensured by data encryption. For encryption of data, symmetric cryptography is used. The main quality of the system is that all cryptographic operations are performed on the client side, which gives users more control on the security of their data, and thus the data are not dependent on the security solutions provided by the servers.*

KEYWORDS: Encryption, Digital Signature Algorithm (DSA)**,** Amazon, Web Identity, Cloud Storage Systems, Cryptography

INTRODUCTION

The term information technology is not so old, but cannot deny its extremely fast growth, especially in the last decade. There is no doubt about the big progress of the internet, which is the main factor in IT world, especially with regard to speed in data transfer, both in terms of wired and wireless communication. People run their business, do their researches, complete their studies, etc. by using the facilities available through the use of internet. All in all, the outsourcing of facility management is becoming more and more common.

Since the need for online services is increasing, the extent of services available through the internet, such as online software, platform, storage, etc., is also growing. This leads to formation of a structured provision of services, "called cloud computing", which actually provides a huge amount of computing resources as services through the internet. One of the important services in the cloud is the availability of online storage, called cloud storage.

Cloud computing is a result of gradual development of providing services by forming clusters and grids of computers. The main concern is to provide a large amount of services in a virtualized manner in order to reduce the server sprawl, inefficiencies and high costs. So, in cloud computing the servers that are used to provide services, among others cloud storage, are fully virtualized. This virtualization mechanism makes it possible for cloud storage users to get the specific amount of storage that they need, and thus they are only required to pay for the used storage.

Since this enormous amount of services is available online, the use of distributed systems is growing, and thus this new technology, namely “cloud computing”, is becoming popular. People are moving towards using cloud storages in order to enjoy the advantages, such as flexibility in accessing data from anywhere. People do not need to carry a physical storage device, or use the same computer to store and retrieve their data. By using cloud storage services, people can share their data with others, and perform their cooperative tasks together without the need of meeting each other so often. Since the speed of data transfer over the internet is increasing, there is no problem in storing and sharing large data in the cloud.

Cloud storage systems vary a lot in terms of functionality and size. Some of the cloud storage systems have a narrow area to focus on, like only storing pictures or e-mail messages. Others provide storage for all types of data. According to the amount of services they provide, they range from being a group of small operations to containing very large amount of services, such that the physical machinery can take up a big warehouse. The facility that houses a cloud storage system is called a data centre. If just one data server is available, and connects it to the internet, it is actually enough to provide a cloud storage system, though it is the most basic level. The common cloud storage systems in the market are based on the same principle, but there are hundreds of data servers that lie at the back end. The computers usually need to be maintained or repaired, so it is important to have copies of the same data on multiple machines. Without this mechanism, a cloud storage system cannot ensure data availability to the clients. Most systems store copies of the data to the different servers that are supplied with different control resources. In this technique, the data would still be available when power failure occurs on one server.

When discussing these improvements, one important issue in IT that must be taken care of is ensuring security. Users use the cloud storage facility to store and share their data, and especially when these data are clandestine, the need of security is compulsory. It means that the confidentiality and integrity of data are needed to be ensured. Furthermore, the stored data must always be available for retrieval that is the system has to provide availability of data. In short, having security in cloud storage is actually ensuring confidentially, integrity and availability of stored data.

Many cloud storage providers claim that they provide a very solid security to their users, but it should be kept in mind that every broken security system was thought once to be unbreakable. A case in point as examples are Google’s Gmail collapse in Europe in February 2009 (Google, 2009), a phishing attack on Salesforce.com in November 2007 (Salesforce.com) and a serious security glitch on Dropbox in June 2011 (Dropbox/8301-31921_3-20072755-28). Looking a bit deeper in the structure of cloud computing systems makes it feel even more insecure, because they make use of multi-tenancy

There are various file systems available for different operating systems, but the main principle is almost the same. Generally, the basic operations that can be performed on files in a file system, are reading a file, writing to a file, deleting an existing file and creating a new file. So, a user can perform read/write operations on his files through a file system, but should it be possible for other users to run the same actions on the user’s files? When this question is raised, the term access control comes to mind. Some kind of access control mechanism must be available, so that a user can assign access limitations to his data. In such a system, a user would be able to grant trusted users read access permission, or both read and write access permission to his data. Besides this, when a user reads his file, he must be assured that the content of his data has not been modified by some unauthorized users, which means that the integrity of data must be guaranteed. The same requirements are applicable when users store their data to online storages. To put it briefly, there is the need to secure stored data.

The term security for data has always been an important issue, and its degree of importance depends on how secret data are. Also, in the olden days, when technology did not exist, people had secret data, and they also controlled the access to their data in some degree. In the digital world, when data are stored to servers; according to the degree of secrecy, the term access control to data is defined more clearly. There are many types of access control mechanisms in different systems, but the main idea is controlling read and write access, which fall under confidentiality, and besides that, ensuring data integrity. Finally, it is also important that the stored data is always available, but it is solely the task of the server to provide availability for the data. To sum up, there are three levels of access permission to the stored data:

i. Verifying the integrity of the stored data.

ii. Verification and read access to the stored data.

iii. Verification, read and write access to the stored data.

To achieve a system that covers the above specifications for access control, cryptography could be used. By using cryptography, operations could be performed locally on the client side, which additionally increases the security level. For data confidentiality, symmetric encryption can be used, and for data integrity, asymmetric encryption can be used. Among the biggest commercial providers of cloud computing are Amazon, Google and Microsoft. Many providers, including the three mentioned providers, have their definitions of cloud computing.

Empirical Basis of the study

NIST has a definition, which sounds: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (example networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models, (Mell and Grance 2009).

NIST has introduced a broad definition, which contains most of the definitions given by cloud computing providers. As mentioned in the definition, cloud computing is typically known by five essential characteristics. These characteristics are on-demand self-service, broad network access, customizability, elasticity and per-usage metering and billing.

The four deployment models are private cloud, community cloud, public cloud and hybrid cloud. These deployment models talk about the different usage of cloud computing. For instance, a private cloud is obviously “smaller” than a hybrid cloud, because a private cloud would provide fewer services that is only those services that are needed in a single organisation.

The services provided by cloud computing are mainly of three types/layers, namely infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS).

These levels can be viewed as a layered architecture, such that services in a higher layer can be composed from the services of the underlying layer. Here IaaS is the lowest layer, and SaaS is the highest.

The following are the services and its contents of each service.

SaaS (Software as a Service) this is the top layer of cloud computing systems, and the services provided here can be accessed through user clients, which can be Ib browsers. Users can use the available software without thinking about where they are installed, and which computing resources they use, and thus it minimises the users' task with regard to software maintenance.

Some examples of software services are Accounting, customer relationship management (CRM), content management (CM), Office suites, video processing, etc.

PaaS (Platform as a Service*)* this is the programmable layer of cloud computing systems. It contains an environment for developing and deploying software. Users do not need to consider about the computing resources and amount of memory that the software would use. PaaS is the middle layer, and it makes operating system, database and web server. In short, it provides a computing platform for end users. Programming languages, Application development tools, Database, Ib server, etc. are some examples of platform as a service.

IaaS (Infrastructure as a Service) is the bottom layer of cloud computing systems is a model in which an organization outsources the equipment used to support operations such as storage, hardware equipment, servers and networking components. The service provider owns the equipment and is responsible for housing, running and maintaining it. The client typically pays on a per-use basis. It can be seen that cloud storage is one of the many facilities cloud-computing systems provide, and it belongs to IaaS layer (Gartner; Yankee 2009).

As mentioned in earlier, the main concern of this research is providing security for cloud storage. In the following, the well-known cloud storage providers mentioned, and their security solutions compared.

Cloud Storage

Cloud storage is an online virtual distributed storage provided by cloud computing vendors. Cloud storage services can be accessed by a web service interface, or a web-based user-interface. One of the advantages is its elasticity. Customers get the storage they need, and they only pay for their usage. By using cloud storages, small organisations save the complexity and cost of installing their own storage devices. The same as cloud computing, cloud storage has also the properties of being agile, scalable, elastic and multi-tenant.

Amazon S3

Amazon, called Amazon Simple Storage Service (Amazon S3), provides one of the Well-known cloud storages. It provides data storage and retrieval by means of web services interfaces, such as REST, SOAP and BitTorrent. Amazon S3 is a key/value store, and it is suitable for storing large files that is up to five (5) Terabytes of data. For storing smaller data, it is more suitable to use Amazon's other data storage, called SimpleDB.

For managing files in large data stores, like cloud storages, relational database systems are not applicable. It would get very complex and almost impossible to use MySQL, for instance, for managing data. Therefore, Amazon S3, SimpleDB and also other cloud storages usually use NoSQL database solutions.

To reduce complexity, Amazon S3 has purposely minimal functionality, so data can only be written, read and deleted. Every object/file is stored in a bucket and retrieved via a unique key. It supports storing 1 byte to 5 terabytes of data, and the number of files to be stored is unlimited.

Security of Amazon S3

Amazon S3 provides security mechanisms, by which a user controls who can access his stored data, and how, when and where the data can be accessed. In order to achieve this security, Amazon S3 provides four types of access control mechanisms:

- "Identity and Access Management (IAM) policies" make it possible to create multiple users under a single AWS (Amazon web Services) account. By using this mechanism, each user can control other user's access to his buckets or files.
- "Access Control Lists (ACLs)" make it possible for a user to grant specific permissions on every file in a selective way.
- "Bucket policies" are used to grant or deny permissions on some or all of objects within a bucket.
- "Query string authentication" is used to share objects through URLs.

Besides these mechanisms, users can store/retrieve data by using SSL encryption via HTTPS protocol. Amazon S3 also provides encryption of data by a mechanism called Server-Side Encryption (SSE). By using SSE, data are encrypted during the upload process and decrypted when downloaded. Users can request encrypted storage, and Amazon S3 SSE handles all encryption, decryption and key management processes. When a user *PUT*s a file and request encryption, the server generates a unique key, encrypts the file using the key, and then encrypts the key using a master key. For ensuring more protection, keys are stored in hosts that are distinct from those, where the data are stored. The decryption process is also performed on the server, so when a user *GET*s his encrypted data, the server fetches and decrypts the key, and then uses it to decrypt the data. The encryption is done by using AES-256, Schneier (2009), Gilbert and Peyrin (2009).

All of the above-mentioned access control mechanisms are server centric, and users have no choice other than trusting Amazon S3.

Google Cloud Storage

Google Cloud Storage is a service for developers to write and read data in Google's cloud. Besides data storage, users are provided with direct access to Google's networking infrastructure, authentication and sharing mechanisms. Google Cloud Storage is accessible via its REST API or by using other tools provided by Google.

Google Cloud Storage provides high capacity and scalability, i.e. it supports storing terabytes of files and large number of buckets per account. It also provides strong data consistency, which means that after uploading your data successfully, you can immediately access, delete or get its metadata. For non-developer users, who require fewer services, Google offers another data storage, called Google Docs, which supports storing up to 1 GB of files.

Google Cloud Storage uses ACLs for controlling access to the objects and buckets. Every time a user requests to perform an action on an object, the ACL belonging to that object determines whether the requested action should be allowed or denied.

Dropbox

Dropbox is a file hosting service that allows users to store and share their data across the internet. It makes use of file synchronisation for sharing files and folders between users' devices. Two MIT students, Drew Houston and Arash Ferdowsi in 2007, founded it and now it has more than 50 million users across the world. Users can get 2GB of free storage, and up to 1TB of paid storage. Dropbox provides user clients for many operating systems on desktop machines, such as Microsoft Windows, Mac OS X and Linux, and also on mobile devices, such as Android, Windows Phone 7, iPhone, iPad, WebOS and BlackBerry. However, users can also access their data through a Ib-based client when no local clients are installed. (About Dropbox) (What are the system requirements to run Dropbox?,),

Dropbox can be used as data storage, but the main focus is file sharing. If a Dropbox client is installed on users' devices, besides storing the shared data on the server side, these data are also stored on shared users' local devices. Whenever a user modifies the shared data on his client, the shared data on the server and on all the other shared clients are also updated (when syncing) according to the performed modification. Dropbox supports revision control mechanism, so users can go back and restore old versions of their files. It keeps changes for the last 30 days as default, but they offer a paid option for unlimited version history. In order to economise on bandwidth and time, the version history makes use of delta encoding, i.e. when modifying a file, only the modified parts of the file are uploaded, Levy and Ari (2010)

Dropbox makes use of Amazon's cloud storage, namely Amazon S3, as their data storage. However, the founder of Dropbox, Drew Houston, has mentioned in an interview (www.cloudnewsdaily.com) that they may build their own data centre in the future. They claim that Dropbox has a solid security for users' data, and they use the same security solutions as banks. For synchronisation, Dropbox uses SSL file transfer protocol, and the stored data are encrypted at the server side using AES-256 encryption. (www.dropbox.com)

Cloud Storage Security Requirements

In the process of storing data to the cloud, and retrieving data back from the cloud, there are mainly three elements that are involved, namely the client, the server and the communication between them. In order for the data to have the necessary security, all three elements must have a solid security. For the client, it is mostly every user's responsibility to make sure that no unauthorised party can access his machine. When talking about security for cloud storage, the security for the two remaining elements is our main concern. On the server side, data must have confidentiality, integrity and availability. Confidentiality and integrity of data can be ensured both on the server side and on the client side. At the end, when introducing the cryptographic access control mechanism, there will be discussion on the differences between server side and client-side security solutions. The availability of data can only be ensured on the server side, so it is the responsibility of the server to make sure that data is always available for retrieval.

Last but not least, the communication between client and server must be performed through a secure channel, i.e. the data must have confidentiality and integrity during its transfer between server and client. One of the ways to achieve secure communication is having a cryptographic protocol, such as SSL.

Cloud Storage Security Solutions

The two mentioned commercial cloud storage providers, Amazon and Google, are large and Well-known providers in the market. Dropbox, as being a cloud storage provider and file sharing service, is also getting more and more popular. Moreover, many other cloud storage providers use various security mechanisms including cryptography. In the following there will be a mention of some of the security solutions that have been suggested or used, and a comparison between these approaches will also be mentioned.

For some types of data, for instance, the data in a digital library, the integrity of data is the main concern, but the confidentiality of data is not relevant. In this case it is important to have a fast mechanism and not so complex communication to verify the integrity of data. For achieving this goal, two approaches are proposed, which are stated in a research work Pascal, (1999). One is called Proof of Retrievability and Uploadability Schemes (PORU), which is a challenge-response protocol used by a cloud storage provider in order to show the client that his data is retrievable without any loss or corruption. The second approach is called Provable Data Possession Schemes (PDP), which is also a challenge-response protocol, but it is weaker than PORU, because it does not guarantee the retrievability and uploadability of data. These two approaches are reasonably fast processes, because the data retrievability and uploadability verified without re-downloading the data. To many other types of users, confidentiality of their data is of much importance. Consequently, many of the commercial cloud storage providers give confidentiality solutions to the clients. The content of table 1 is taken from Virvilis and Heidneberg (2011), which contains security comparisons between popular commercial cloud storage providers. (The last row containing information about Dropbox is not stated in the paper. *The information is taken from Dropbox's website, and it has been added to the table).*

Table 1: A Comparison Between Two Cloud Storage Solutions.

Cloud Solutions	Own Data Center	Sync	Secure transmission	Data Encryption	Multifactor Authentication	Free Space
Amazon S3	Yes	-	-	No	No	-
Azure	Yes	-	-	No	No	-
Carbonite	Yes	No	Yes	Blowfish-128 Symmetric at client side	Yes	-
Mozy	Yes	Yes*	Yes	AES-256 or Blowfish-448 Symmetric at client side	No	2GB
Spideroak	Yes	Fes	Yes	AES-256 Symmetric at client side	No	2GB
SugarSync	Yes	Yes	Yes	AES-128	No	5GB
Ubuntu One	No	Yes	Yes	No	NO	2GB

Wuala	Yes	Yes	Yes	AES-128 Symmetric at client side	No	1GB
Dropbox	No	Yes	Yes	AES-256 Symmetric at client side	No	2GB

Source Virvilis and Heidelberg (2011)

Table 1 also compares other features of cloud storages, like whether or not they have their own data centers, whether they support syncing between multiple computers or not, and etc., but the column "Data encryption" is relevant here. It shows that six of the mentioned cloud solutions support data confidentiality in form of symmetric data encryption, and four of them support this mechanism on the client side. (However, Amazon S3 provides SSE as mentioned before, but since SSE is a new addition to Amazon S3, it is not mentioned in the table.) It could be seen that ensuring integrity of data is missing in these cloud solutions. These two approaches, POR and PDP, for verifying data integrity are already described, but as mentioned, the two approaches are proofs for showing irretrievability of the data without downloading. It is suitable for systems with large data that does not need to be secret. Once the integrity of the whole data is ensured, one can read some amount of the data that he needs.

Cryptography

Cryptography is the most common technique for ensuring a secure communication between two parts in the presence of a third party. If M (Martha) and S (Sammy) send messages to each other, and they do not want others to read or change the content of their messages, then it means that they want to have a secure communication. In this communication, a broadcast medium *T* is used, a message to S by means that M sends of *T*. A third party, who wants to interfere this communication by accessing/altering the message, is called an intruder *I* whenever a message is on its way towards the destination, it is in danger of being accessed by *I*, who can perform the following actions:

i. Message can be *blocked*, so it never reaches its destination, and thus the availability is violated.

ii. Message can be *intercepted*, so it is not secret anymore, and thereby the confidentiality is broken.

iii. The content of the message can be *changed*, and by that the integrity is violated.

iv. Message can *faked* and sender *A* impersonated, and send the message to *B*. This violates also the integrity of the message.

In communications between two parts, the security of messages can be exposed to the above four dangers. In cryptography, the encryption techniques are used to handle all these security issues. Encryption is actually the most important method to insure security in communications.

Encryption and Decryption

The techniques used in cryptography are encryption and decryption of data. It can be called encoding and decoding, or enciphering and deciphering. Encryption, encode or encipher is a technique by which the original text, often called the plaintext, is changed, such that the meaning of the text is hidden, that is the plaintext is transformed into an unintelligible or incomprehensible string of text, often called the ciphertext. In order to change the ciphertext back to the plaintext, it has to be decrypted, decoded or deciphered.

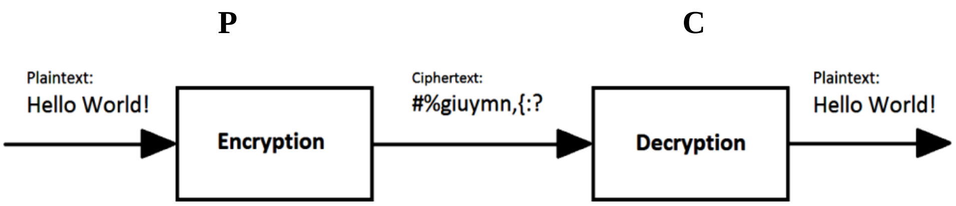


Figure 1: Encryption/Decryption

In Figure 1, the plaintext P is considered as a sequence of characters P = <H,e,l,l,o, ,W,o,r,l,d,!> and in the same way the ciphertext C = <#,%,g,i,u,y,m,n,,,{,:,?>. A system that encrypts and decrypts data is called a cryptosystem. If the two processes in a cryptosystem are to be denoted formally, it would be = $E(P)$ and $P = D(C)$, where C is the ciphertext, P is the plaintext and E and D are encryption and decryption algorithms respectively. The cryptosystem is denoted as $P = D(E(P))$, which means that the plaintext P is the decryption of encrypted P. In cryptosystems, a key K is usually used with an algorithm in order to encrypt or decrypt the data. Whenever a key is used for both encryption and decryption, then the process is called symmetric encryption, and the key is called symmetric key. In this case, the encryption and decryption algorithms are symmetric and they can be considered as reverse operations with regard to each other. The official notations would be $C = E\ (K, P)$ and $P = D\ (K, C)$, and the cryptosystem is denoted as $P = D\ (K, E\ (K, P))$. If the key used for encryption/decryption is not the same, then the process is called asymmetric encryption. Here two keys are used, namely an encryption key (often called private key), KE for encryption and a decryption key (often called public key), KD for decryption. The official or formal notations in this case would be $C = E(KE, P)$ and $P = D(KD, C)$ and the cryptosystem is accordingly denoted as $P = D(KD, E(KE, P))$. Figure 1 and Figure 2 shows overviews of the two encryption/decryption methods.

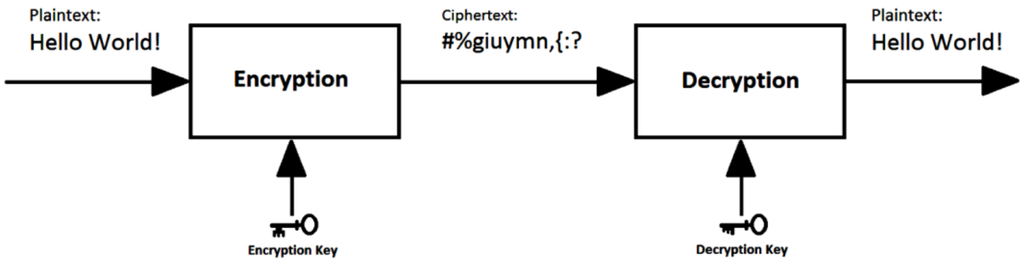


Figure 2 : Encryption/Decryption using Different Keys

Symmetric Algorithms

Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are two most famous or well-known symmetric algorithms. Another symmetric algorithm, which is a public domain algorithm, is called Blowfish.

Data Encryption Standard (DES)

Data Encryption Standard (DES) is one of the famous symmetric algorithms. It is a block cipher with the block size of 64 bits. It was the result of a research set up by International Business Machines (IBM) Corporation in the late 1960's which resulted in a cipher known as LUCIFER. In the early 1970's it was decided to commercialize LUCIFER and a number of significant changes were introduced. IBM was not the only one involved in these changes as they sought technical advice from the National Security Agency (NSA) (other outside consultants were involved but it is likely that the NSA were the major contributors from a technical point of view). The altered version of LUCIFER was put forward as a proposal for the new national encryption standard requested by the National Bureau of Standards (NBS).

According to Shah (2012), the DES was published by the United States' National Bureau of Standards in January 1977 as an algorithm to be used for unclassified data (information not concerned with national security).

Mandar (2013), affirm that from 2001 the AES will replace DES. After 25 years of analysis, the only security problem with DES found is that its key length is too short. DES uses a 56-bit key which can be broken using brute force methods, & is now considered to be insecure for many applications. It was acknowledged that DES was not secure as a result of advancement in processing power computer. From (Sharma, 2010), the purpose of NIST was to define a replacement for DES that can be used in non-military information security applications by US government agencies.

The Data Encryption Standard (DES), as specified in FIPS Publication 46-3, is a block cipher operating on 64-bit data blocks. It is a block cipher with key length 56 bits and it had been used as a standard for encryption until 2000. The effective key length of the DES is 56 binary digits (bits) and the straightforward "work factor" of the algorithm is 2^{56} (i.e., the number of keys that would have to be tried is 2^{56} or approximately 7.6 x 10^{16}). Hellman and Diffie argued that, in certain situations, a symmetric characteristic of the algorithm would cut this number in half and that on the average, only half of these would have to be tried to find the correct key. They also noted that increasing the key length by 8 bits would "appear to outstrip even the intelligence agencies' budgets" but that "decreasing the key size by 8 bits would decrease the cost making the system vulnerable to attack by almost any reasonable sized organization." It was thus argued that the length of the key was critical to the maximum security provided by the proposed standard (Hellman, 1977). DES uses eight predefined S-boxes which have been determined by the U.S. National Security Agency (NSA). Using the S-boxes, groups of six bits are mapped to groups of four bits. These S-boxes are resistant against an attack called differential cryptanalysis, which was first known in the 1990s. Largely, the encryption process of DES is done in 16 rounds.

On the process of DES encryption, a message:

i. The input key is used to derive sixteen 48-bit keys (known as subkeys). Each of these keys is then used in each round.

ii. The right half is expanded from 32 bits to 48 bits using another fixed table.

iii. The result is combined with the subkey for that round using the XOR operation.

iv. Using the S-boxes the 48 resulting bits are then transformed again to 32 bits, which are subsequently permutated again using yet another fixed table. In the next round, this combination is used as the new left half.

Yogesh (2011), stated that the DES algorithm is vulnerable to Linear Cryptanalysis attacks. By such an attack, the algorithm in its sixteen rounds can be broken using 2^{43} plaintexts. This vulnerability raises a notable risk when encrypting bulk data that may be predictable with keys that are constant.

Biham and Shamir (1991) presented a differential attack, by which a key can be recovered in 2^{37} time using 2^{37} cipher texts taken from a pool after encrypting 2^{47}. According to Vikendra (2013) DES algorithm is a 64-bit block cipher which means that it encrypts data of 64 bits at a time.

METHODOLOGY

In implementation, Java programming language, Java Cryptography Architecture (JCA) package, and Java Cryptography Extension (JCE) are used. JCA is a core Application Programming Interface (API) of the Java programming language and is designed to allow developers to incorporate both low-level and high-level security functionalities into their programs**.** The purpose of this system is to put up a security solution, namely cryptographic access control mechanism, into practice. As desired, the solution is intended to be applied to a cloud storage system, Technology is bound to catch up to all cryptosystems and beat their computational limits. For this reason, any new encryption method should be welcomed as future input to viable alternatives, especially suggestions that comply to the “low computational cost”-“high flexibility to cryptanalysis” paradigm.

This new proposed algorithm is a block cipher that divides data into blocks of equal length and then encrypts each block using a special mathematical set of functions known as Key. This algorithm uses variable key length, which will vary from 65 bytes to 72 bytes of symmetric key technique for encryption and decryption of data that is; it uses the same key at both ends. Selection of the key is purely random based. Thus, the key distribution predicament can be handled easily. Another positive point of the algorithm is that it protects the cipher text from Brute-force attacks, as the key is lengthy in the encryption process because of 2^{288} required to break the key.

Cryptographic Models

The study is designed on the basis of two models:

I. Simplified Model

II. Conventional Model

Simplified Model

Diagrammatically, these models that formed the basis of the design of the work are shown in figure 4.

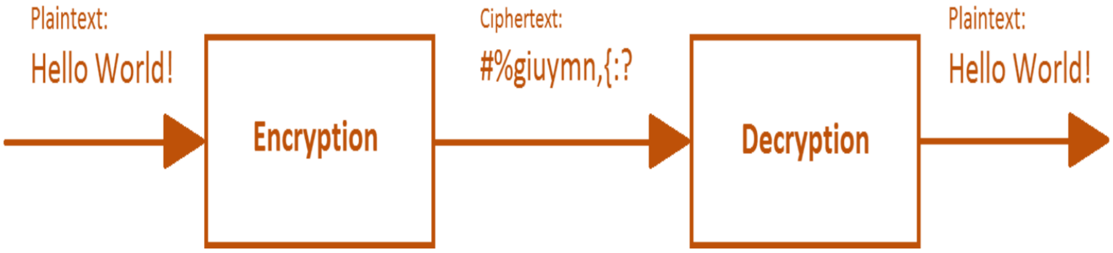


Figure 4: Simplified Cryptographic Model

Figure 4 depicts a simple mode of data encryption. The plaintext (Hello World) is encrypted into the ciphertext (#%giuymn,{:?) and then decrypted into the plaintext at the other point.

Conventional Cryptographic Model

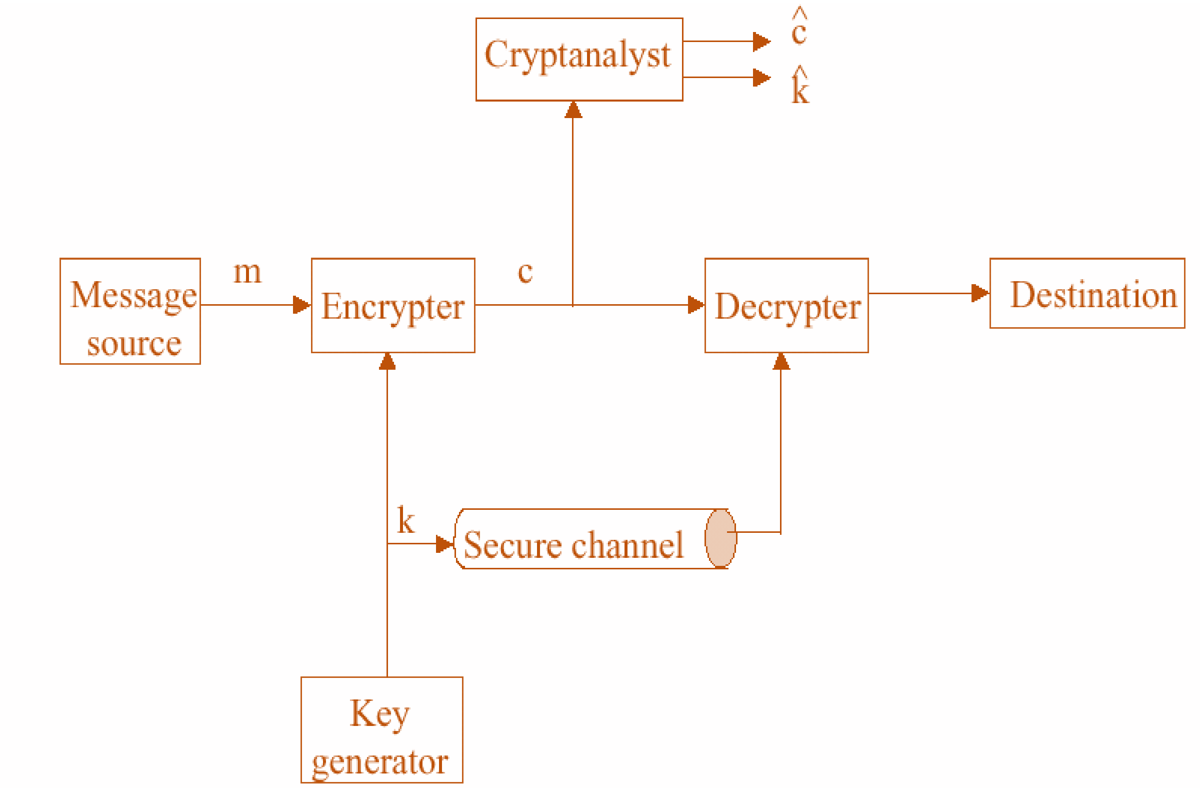


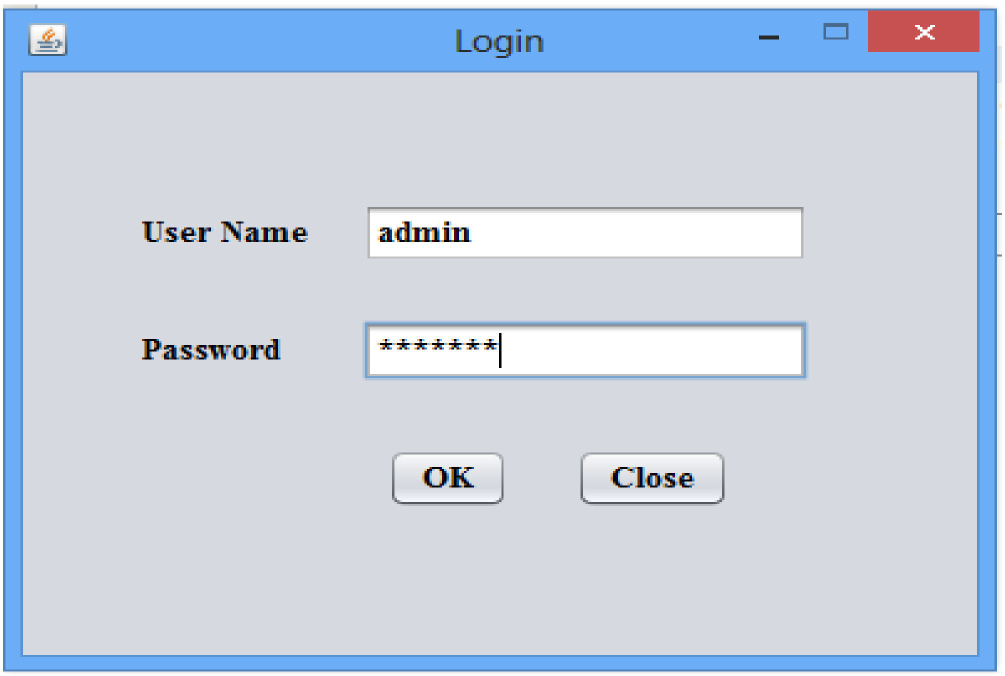
Figure 5: Conventional Cryptographic Model

In figure 5, the conventional model of data encryption is depicted. And with this model, data originates from source, passes through the encrypter, the cryptanalyst and a key is generated for its decryption at its final destination

ANALYSIS AND DISCUSSIONS OF RESULTS

Interface of the System

Per the design of the system modules, the user is supposed to log in as an authenticted user through the user interfcae before getting access to other modules of the aplication



User Name admin

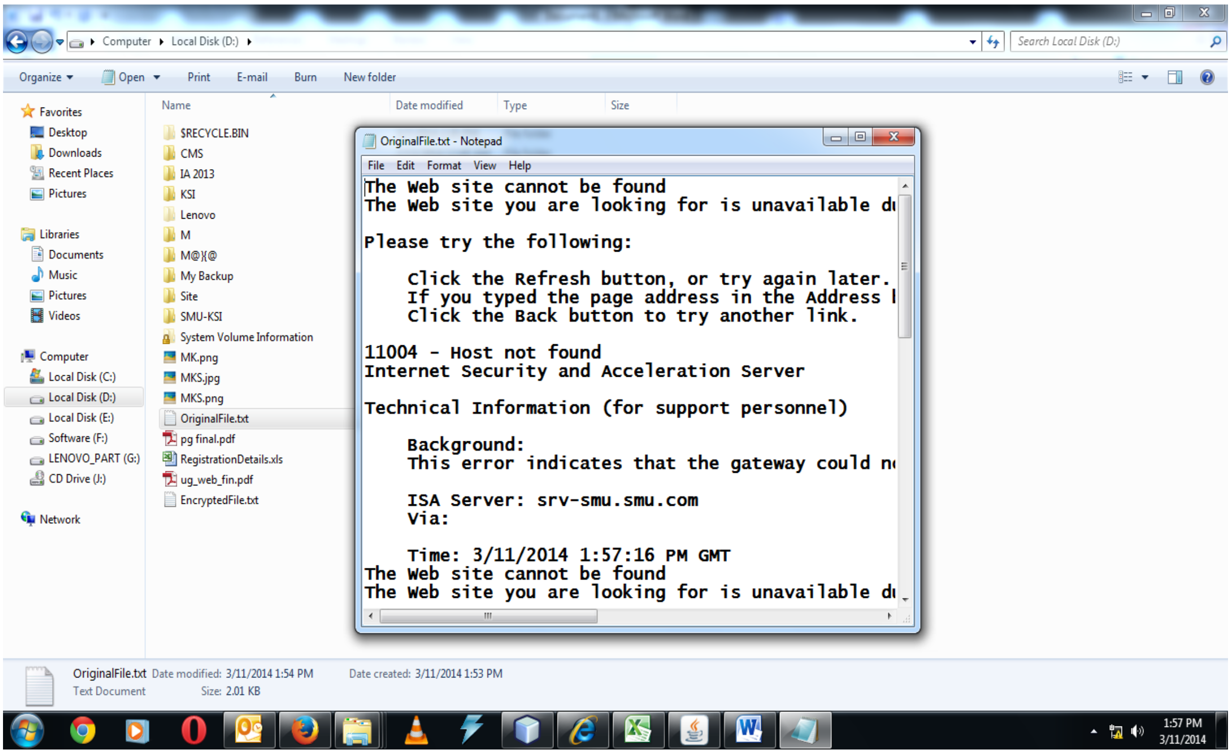
Password *******

Figure 8: Authentication Interface for Encryption and Decryption

In figure 8, the user is supposed to perform the following actions before getting access into the system

- ✓ Enter your username and password in the textboxes provided
- ✓ Click the Ok button to initiate the user authentication process
- ✓ Use the Cancel button to exit the application
- ✓ Module for encryption and Decryption

Source file in a plaintext that could be loaded for encryption in the module displayed



```
The Web site cannot be found
The Web site you are looking for is unavailable du

Please try the following:

    Click the Refresh button, or try again later.
    If you typed the page address in the Address l
    Click the Back button to try another link.

11004 - Host not found
Internet Security and Acceleration Server

Technical Information (for support personnel)

    Background:
    This error indicates that the gateway could no

    ISA Server: srv-smu.smu.com
    Via:

    Time: 3/11/2014 1:57:16 PM GMT
The Web site cannot be found
The Web site you are looking for is unavailable du
```

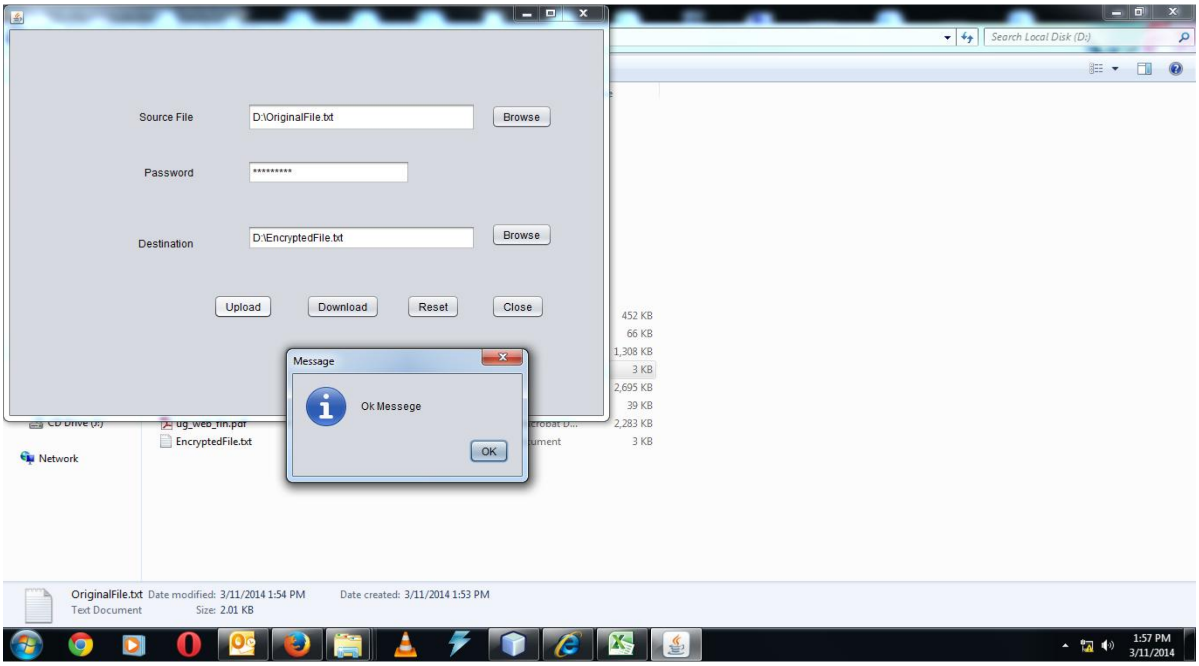
OriginalFile.txt Date modified: 3/11/2014 1:54 PM Date created: 3/11/2014 1:53 PM
Text Document Size: 2.01 KB

Figure 9: Interface for Encryption and Decryption

Figure 9 displays a note pad plaintext for encryption

Module for file uploading and downloading

The system is designed in such a way that the user can upload a file to the cloud service provider after the plaintext is encrypted of download a file from the service provider’s server and decrypt.



Source File: D:\OriginalFile.txt

Password: *********

Destination: D:\EncryptedFile.txt

Message: Ok Messege

OriginalFile.txt Date modified: 3/11/2014 1:54 PM Date created: 3/11/2014 1:53 PM
Text Document Size: 2.01 KB

Figure 10: User Interface Design for File Access

In figure 10:

i. The browse button enables the user to locate a file to be encrypted or decrypted from its resident location

ii. After file selection, the system will have to authenticate the user through password entry

iii. Click the browse button to select destination to place the deciphered file

iv. Click on Decrypt button or encrypt

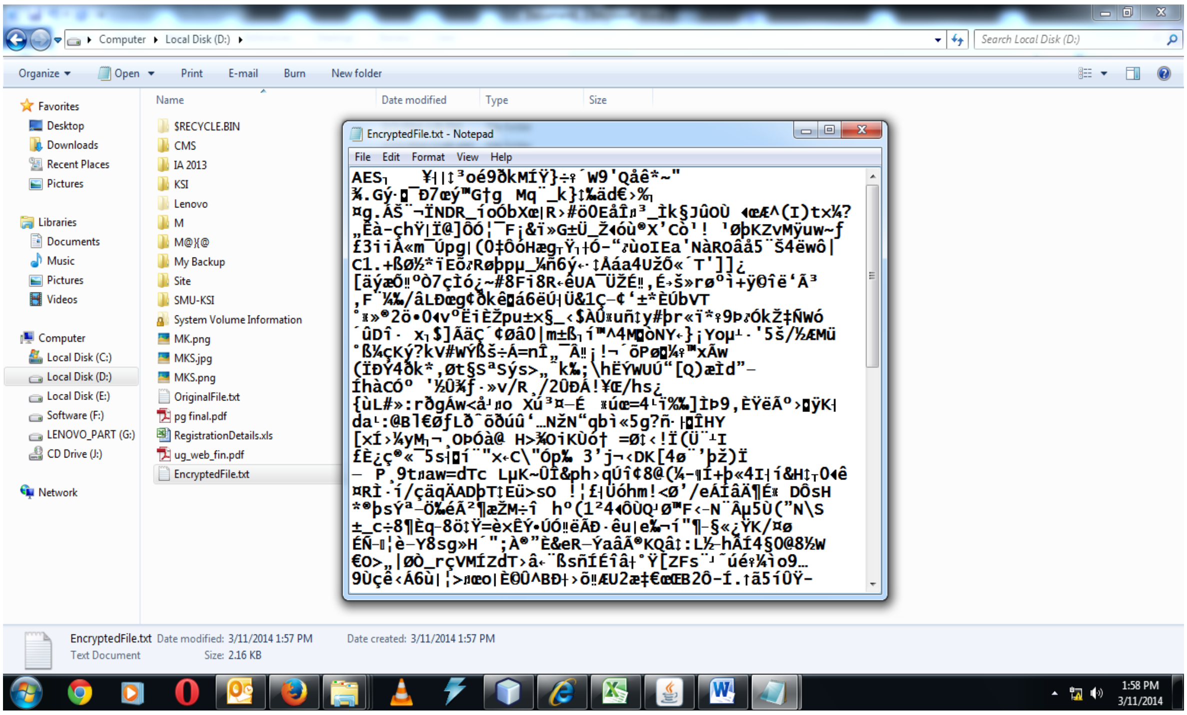
v. The file is then encrypted or decrypted

vi. The upload and download buttons are also provided to enable users to copy files from the cloud server or send files to the server

vii. Finally, the user can reset or exit the application through the quit or reset buttons

Encrypted file display

After the file has been loaded and user authenticated, the file is then encrypted by the system



```
AES┐    ¥┤|↕³oé9ðkMÍŸ}÷♀´W9'Qåê*~"
¾.Gý·▯¯Ð7œý™G†g  Mq¨_k}↕‰äd€›%┐
¤g.ÁŠ¨¬ÏNDR_íoÓbXœ|R›#ö0EåÎ♬³_Ìk§JûOÙ ◄œÆ^(I)tx¼?
„Èà-çhŸ|Ï@]ÔÓ¦¯F¡&ï»G±Ü_Ž◄óù®X'Cò'! 'ØþKZvMÿuw~ƒ
£3iiÀ«m¯Úpg|(0‡ÔóHæg┬Ÿ┐┼Ó-"♪ùoIEa'NàROâå5¨Š4ëwô|
C1.+ßØ½*ïEõ♪Røþpµ_¼ñ6ý←↕Ååá4UžÕ«´T']]¿
[äýæÕ‼ºÒ7çÌó¿~#8Fi8R←êUA¯ÜŽÉ‼,É→š»røºì+ÿ©îë'Ã³
,F¨¼‰/âLÐœg¢ðkê▯á6ëÚ┤Ü&1Ç–¢'±*ÈÚbVT
°☼»®2ö•0◄vºËiÈŽpu±x§_‹$ÀÛ☼uñ↕y#þr«ï*♀9Þ♪ÓkŽ‡ÑWó
´ûDî·  x┐$]ÃäÇ´¢ØâO|m±ß┐í™^4M▯òNY←}¡Yoµ┴·'5š/½ÆMü
°ß¼çKý?kV#WÝßš÷Á=nÎ„¯Â‼¡!¬´õPø▯¼♀™xÃw
(ÏÐÝ4ðk*,øt§Sªsýs>„˜k‰;\hËÝWUÚ"[Q)æÌd"–
ÍhàCÓº '½Û¾ƒ·»v/R¸/2ÛÐÁ!¥Œ/hs¿
{ùL#»:rðgÁw<å┘♬o XúÚ³¤–É ☼úœ=4└ï%‰]ÌÞ9,ÈŸëÃº›▯ÿK┤
da└:@Bl€ØƒLð˜õðúû'…NžN"qbì«5g?ñ·├▯ÎHY
[xÍ›¼yM┐¬¸OÞÓà@ H>¾OiKÙó† =Ø↕‹!Ï(Ü¨┴I
£È¿ç®«¯5s┤▯í¨"x←C\"Óp‰ 3'j¬‹DK[4ø¨'þž)Ï
– P¸9t♬aw=dTc LµK~ÛÏ&ph›qÚî¢8@(¼–¶Í+þ«4I┤í&H↕┬0◄ê
¤RÌ·í/çäqÄADþT↕Eü>sO !¦£┤Úóhm!<Ø'/eÁÏâÄ¶É☼ DÔsH
*®þsÝª–Ö‰éÃ²¶æŽM÷î  hº(1²4◄ÔÙQ┘Ø™F‹–N¨Âµ5Ù("N\S
±_c÷8¶Èq–8ö↕Ÿ=èxÊÝ•ÚÓ‼ëÃÐ·êu|e‰¬í"¶–§«¿ÝK/¤ø
ÉÑ–▯¦è–Y8sg»H´";À®"È&eR–ÝaâÃ®KQâ↕:L½–hÂÍ4§0@8½W
€O>„|ØÒ_rçVMÍZdT›â←¨ßsñÍÉîâ┼°Ÿ[ZFs¨┘˜úé♀¼ìo9…
9Ùçê‹Á6ù|¦>♬œo|È©Û^BÐ┼›õ‼ÆU2æ‡€œŒB2Ô-Í.↑ã5íÛŸ-
```

Figure 11: Interface Display for Encrypted Files

Figure 11 displays a notepad file that has been loaded and encrypted by the application

Here in this phase, there will be evaluation of the implemented system with regard to its performance and security. In the first part, the performance of different functionalities, namely storing data, retrieving data, encryption/decryption, digital signature and key management is evaluated. In the second part, evaluation of the immunity of the system against security will be discussed as against various possible attacks on the security of the system. On the basis of the results achieved from the evaluation, there will be discussion about further improvements that can be applied to the system.

All the tests are run on a laptop with the following specifications:

- OS name: Microsoft Windows 7
- System type: 32 bit
- Processor: Intel(R) Core(TM) 2 Duo CPU, 2,10 GHz
- Physical Memory (RAM): 2,00 GB
- Hard disk drive: 160 GB, 5400 rpm, ATA interface

Performance Evaluation

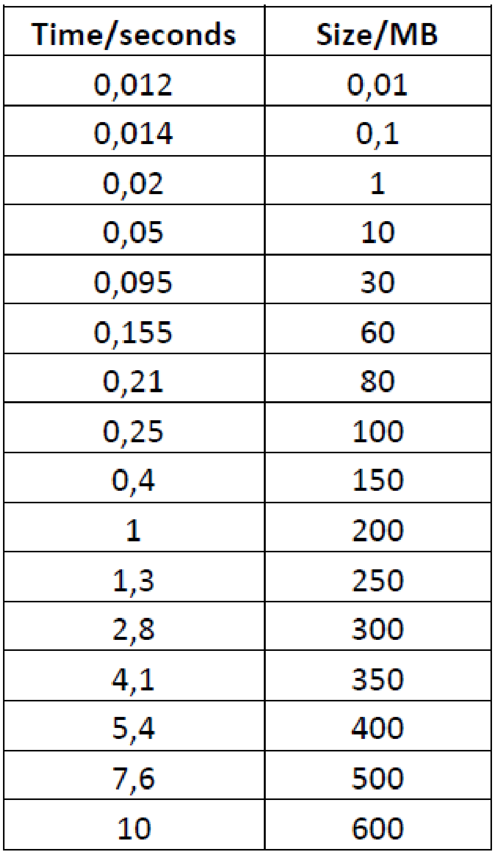
Since security solution is applied on the Infinispan data grid, there will be test on the performance of storing data to and retrieving data from the Infinispan. The test will be done both with and without cryptographic access control.

As known, the system is implemented in Java, so every reading and writing that is applied to a file, is performed by using Java's FileInputStream and FileOutputStream respectively. The two mentioned Java classes are the basis for all readings and writings in our system. So here at the beginning, the speed of copying a file from one place to another place on the hard disk by using both Windows' "copy and paste" function and Java is tested, in order to have a comparison between them.

Comparison between Windows and Java file copying

Table 3 shows the average time that took to copy and paste files with different sizes using Windows' "copy and paste".

Table 3: Comparison Between Windows and Java Copying.



Time/seconds	Size/MB
0,012	0,01
0,014	0,1
0,02	1
0,05	10
0,095	30
0,155	60
0,21	80
0,25	100
0,4	150
1	200
1,3	250
2,8	300
4,1	350
5,4	400
7,6	500
10	600

In order to have an overview of the speed of copy and paste function, A graph for table 3 is drawn, where the average speed is determined.

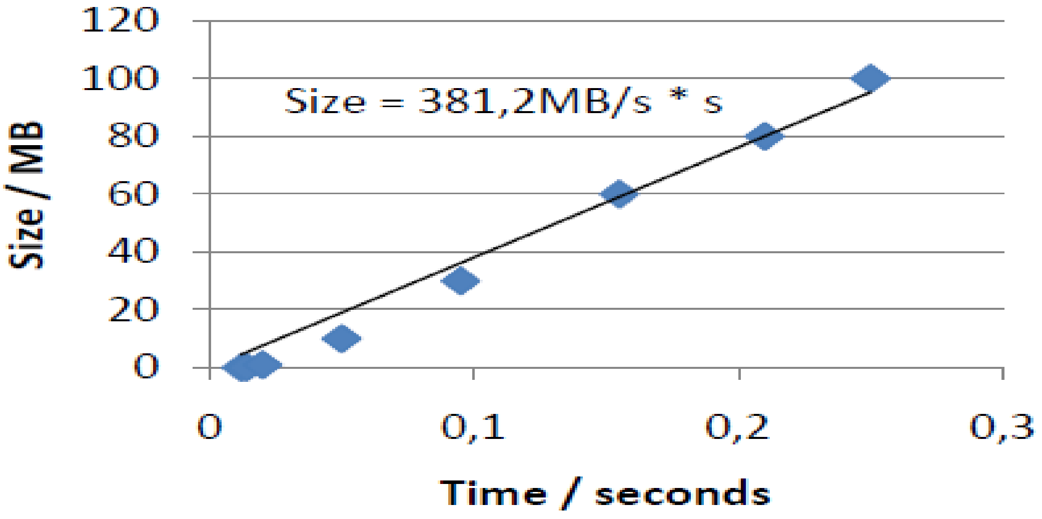


Figure 12 Copying Files up to 100 MB

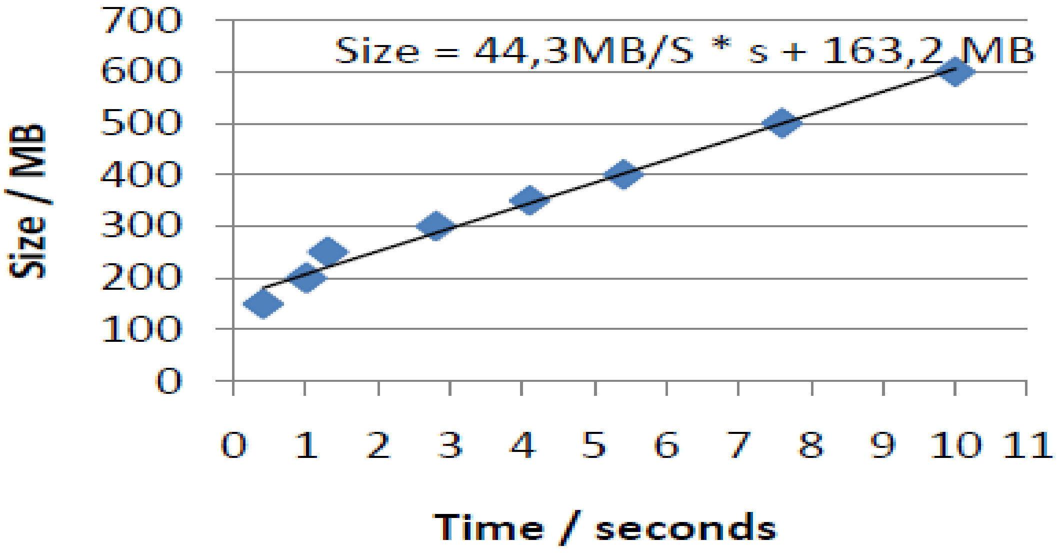


Figure 13 Copying Files, 100 – 600 MB

Figure 12 and Figure 13, shows two graphs belonging to the data in table 3.

Figure 12 contains the graph for files up to 100 MB, and the average speed of copying is approx 381 MB/s. For the files above 100 MB (until 600 MB) the average speed of copying is approx 44 MB/s, so it is much faster to copy small files. The overall average speed is shown in Figure 14.

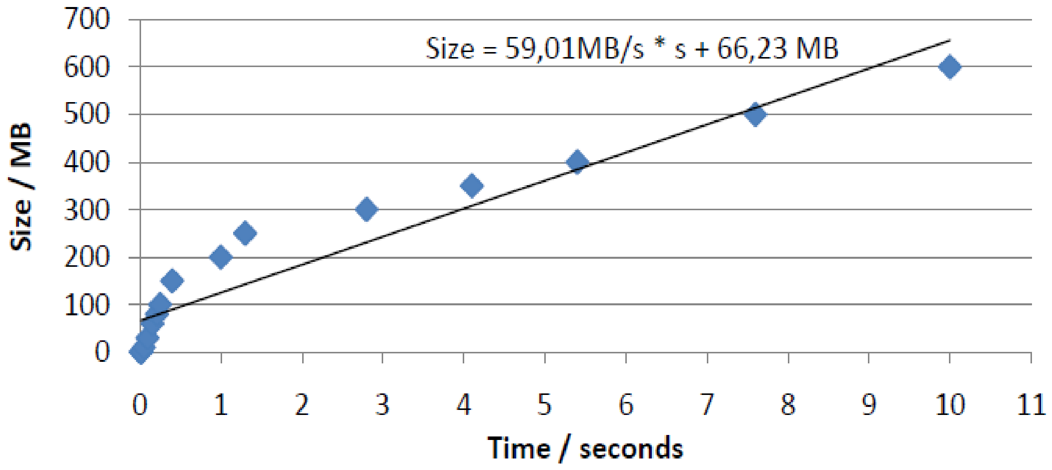


Figure: 14 Copying Files 0,01 MB – 600 MB

As shown in Figure 14, the overall average speed of copying files in Windows is approx 59 MB/s.

Copying Files using Java

The table below shows the average time that took for copying files with different sizes.

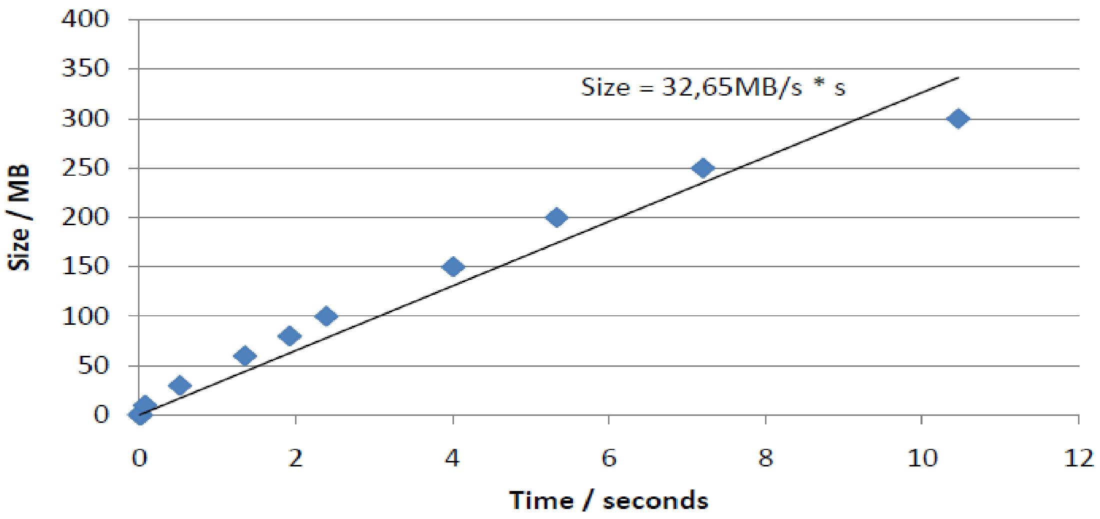


Figure 15: Copying Files using Java, files: 0,01 MB – 600 MB

Figure 15 contains a graph showing the speed of copying files up to 300 MB. It could be seen that the average speed is approximate 32 MB/s. It appears that file copying in Windows is twice as fast as copying files using Java. As a result, the file transferring mechanism in our system is of course affected by this low performance.

Encryption using AES-128

Here the performance of encryption is tested, which is performed by using AES-128.

Table 4: Test Performance of Encryption using AES-128

With key generation & saving the key		Using an existing key	
Time/sec	Size/MB	Time/sec	Size/MB
0,084	0,01	0,004	0,01
0,080	0,1	0,015	0,1
0,110	1	0,048	1
0,400	10	0,330	10
1,100	30	1,075	30
2,065	60	2,034	60
2,670	80	2,650	80
3,295	100	3,250	100

Table 4 shows two tests, where the first one is with generating the symmetric key and saving it to disk, and the other test is performed by using an existing symmetric key. As expected, the encryption is faster when using an existing key, but the difference is not too much, because a symmetric key of length 128 bits is rather small, and thus its generation and saving would not take too much time.

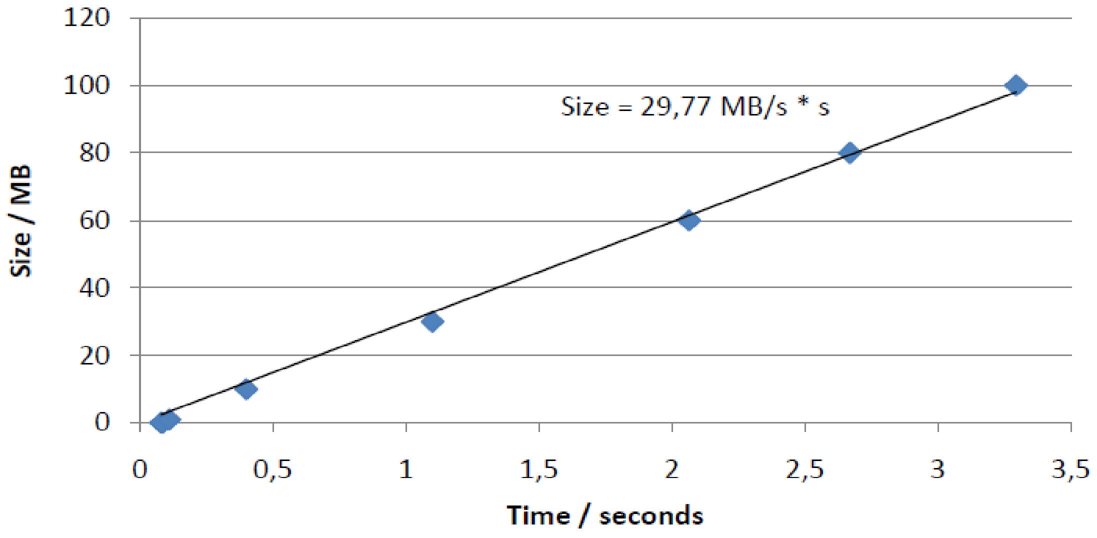


Figure 16: Encryption using AES-128, with key Generation, files: 0, 01 MB – 100 MB

In Figure 16 the average speed of encrypting files with key generation is approximate 30 MB/s.

Table 5: Decryption using AES-128

Time/sec	Size/MB
0,001	0,01
0,005	0,1
0,038	1
0,365	10
1,066	30
2,135	60
2,907	80

A graph for table 5 determining the average speed of decryption is shown in Figure 17.

In Figure 17, the average speed for decryption is approximate 28 MB/s, which is almost the same as encryption speed. As AES uses a symmetric encryption algorithm, so the encryption and decryption processes are the inverse of each other, and as a result their speeds would also be the same.

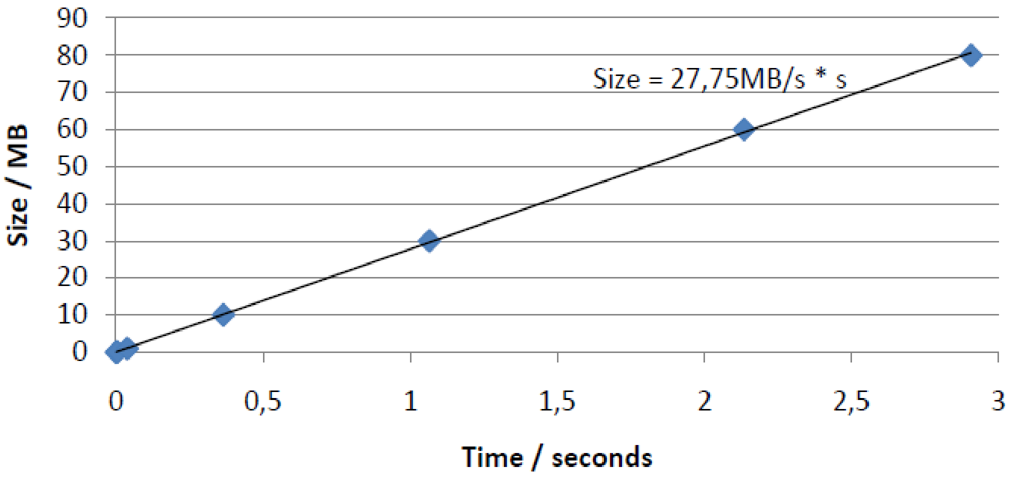
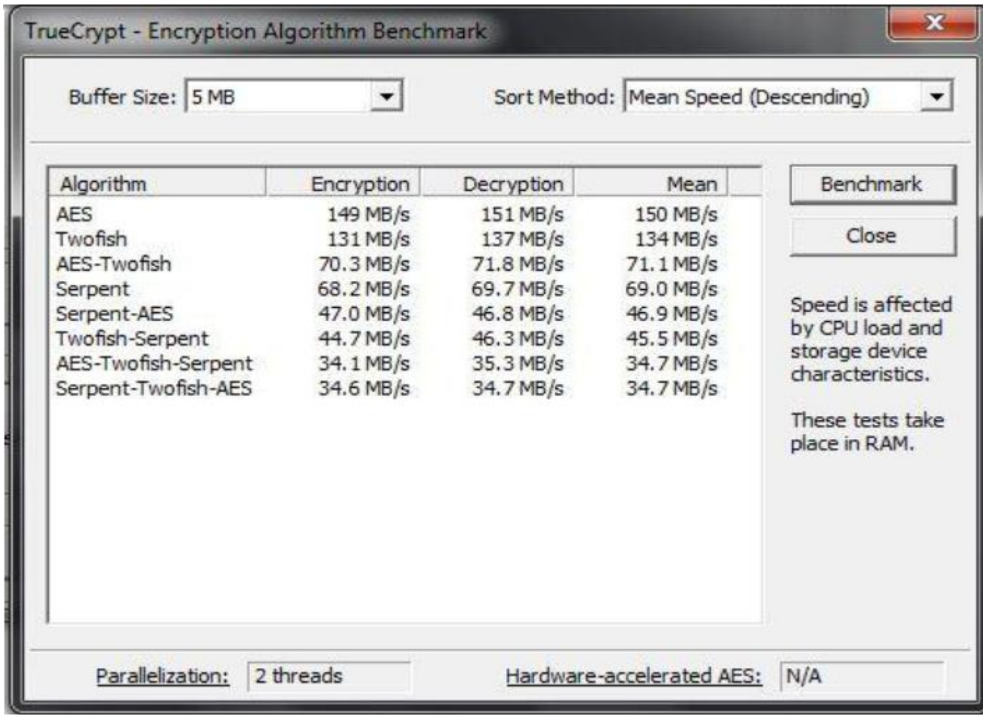


Figure 17: Decryption using AES-128, Files: 0,01 MB – 80 MB

AES is widely used in the world, and there is a lot of software available in the market that provides data encryption using AES. One the most Well-known software is TrueCrypt, which is a freeware. TrueCrypt also contains a benchmark for testing the performance of the supported encryption algorithms. It would be a good idea to compare out implementation of AES, with the TrueCrypt implementation.

TrueCrypt Encryption and Decryption



TrueCrypt - Encryption Algorithm Benchmark

Buffer Size: 5 MB

Sort Method: Mean Speed (Descending)

Algorithm	Encryption	Decryption	Mean
AES	149 MB/s	151 MB/s	150 MB/s
Twofish	131 MB/s	137 MB/s	134 MB/s
AES-Twofish	70.3 MB/s	71.8 MB/s	71.1 MB/s
Serpent	68.2 MB/s	69.7 MB/s	69.0 MB/s
Serpent-AES	47.0 MB/s	46.8 MB/s	46.9 MB/s
Twofish-Serpent	44.7 MB/s	46.3 MB/s	45.5 MB/s
AES-Twofish-Serpent	34.1 MB/s	35.3 MB/s	34.7 MB/s
Serpent-Twofish-AES	34.6 MB/s	34.7 MB/s	34.7 MB/s

Speed is affected by CPU load and storage device characteristics.

These tests take place in RAM.

Parallelization: 2 threads

Hardware-accelerated AES: N/A

Figure 18: TrueCrypt Encryption Algorithm Benchmark

TrueCrypt uses AES-256 as one of the encryption algorithms. Figure 18 shows a benchmark, which is one the features provided in TrueCrypt. After running the Benchmark, it gives an average for the speeds of encryption and decryption for the supported algorithms. The average encryption speed for AES is 149 MB/s, and the decryption speed is 150 MB/s. It is 5 times faster than our implementation of AES. As mentioned in TrueCrypt's Ibsite (www.truecrypt.sourceforge.net), a file container can be created, where encrypted files are stored. When reading/writing the files from the file container, they are encrypted/decrypted on the fly, i.e. portions of data are encrypted/decrypted in RAM, which results in very good encryption performance.

Table 6: Signing data using RSA Signature Scheme with SHA-512

With key pair generation & saving		Using an existing private key	
Time/sec	Size/MB	Time/sec	Size/MB
0,400	0,01	0,027	0,01
0,290	0,1	0,036	0,1
0,372	1	0,080	1
0,800	10	0,540	10
1,860	30	1,515	30
3,215	60	2,977	60
4,163	80	3,950	80
5,177	100	4,810	100

Table 6 shows two different tests for signing data, one is with key pair generation and saving, and the other one is with using an existing private key. As expected, when using an existing private key, the signing process is performed a little faster.

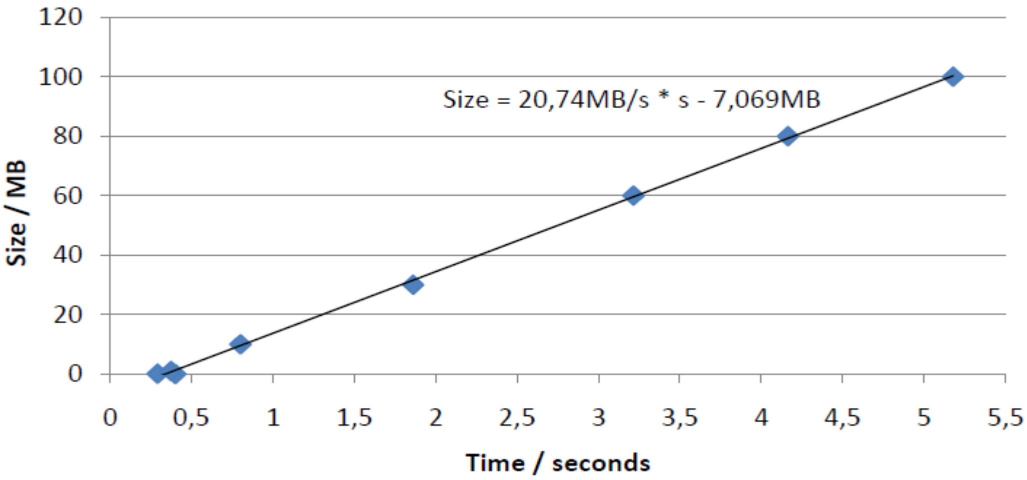
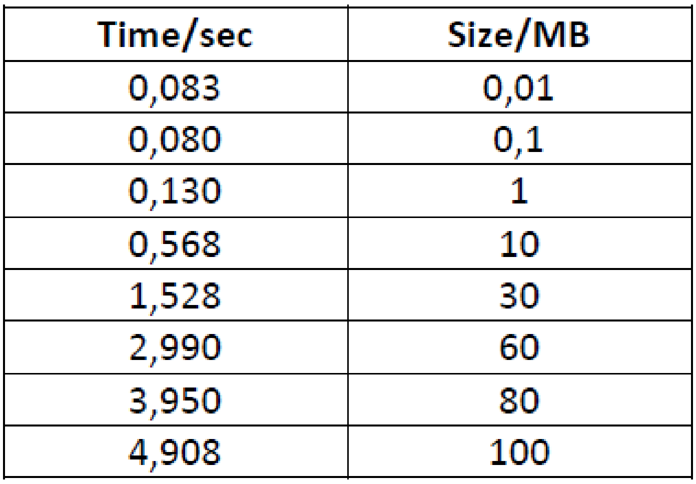


Figure 19: Signing Data using RSA with key pair Generation, files: 0,01 MB – 100 MB

Figure 19 shows that the average speed of signing data is approximate 21 MB/s, which is actually slower than the encryption/decryption process being approximate 30 MB/s. It can be deduced that it is not the actual data that is signed, but a short digest of the data. Since the asymmetric encryption is always much slower than symmetric encryption, it would have taken much more time to sign large files without using a hash function.

Table 7: Verifying data using RSA



Time/sec	Size/MB
0,083	0,01
0,080	0,1
0,130	1
0,568	10
1,528	30
2,990	60
3,950	80
4,908	100

Table 7 shows the time it took for various file sizes to be verified using RSA.

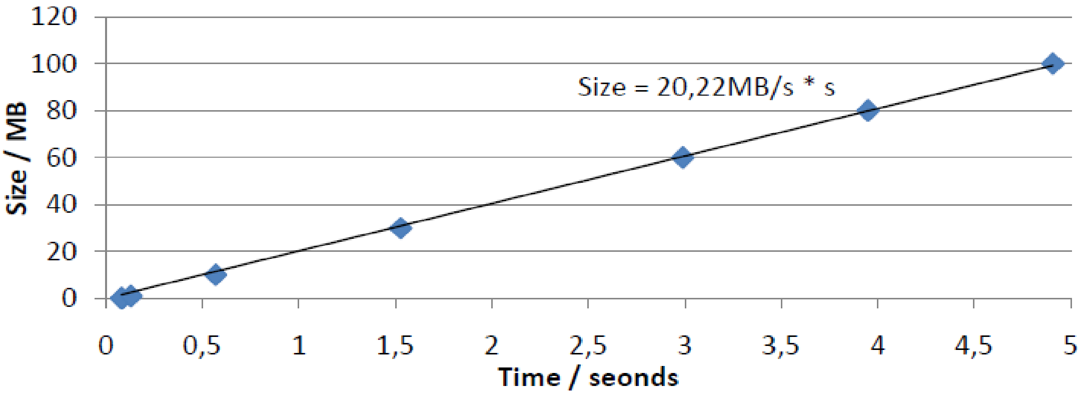


Figure 20: Verifying Data using RSA, Files: 0,01 MB – 100 MB

In Figure 20, the average speed for verifying data is approximate 20 MB/s. It is almost the same as the speed of signing data.

Until now data storage and retrieval have been tested and found out the chunk size for optimal speed, and moreover encryption and signature for different file sizes have also been tested. In the following test on storing and retrieving data with encryption, signature and data persistency will be conducted, in order to have an overall average speed of the system.

Security Evaluation

For granting read & write access, three keys must be read for every file, after which they must be appended to the key ring, whereas for granting read access, two keys are involved for every file, so it will take a little longer to create a key ring with all three keys. Since each key has the same size for all types of data, the size of data does not have any affect in this context. The two first columns in the above table shows the times it took to create key ring for assigning read access to different number of files. The third and fourth columns contain the same data regarding read & write access. As expected, the times in the column related to read access are a little less.

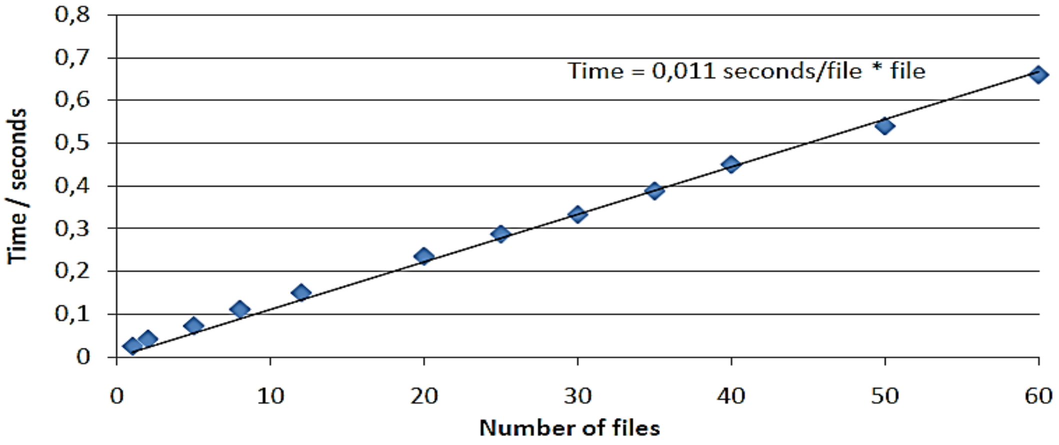


Figure 21: Assigning Read & Write Access to Files

Figure 21 shows a graph for creating key ring when assigning read & write access to a file. It can be seen that in average it takes about 11 milliseconds per file to create a key ring. For example, if 100 files from the data grid are selected, and assign read & write access to them by creating a key ring, then it well approximately takes 11*100 = 1100 milliseconds, or 1, 1 second to create it. So as a result, the process of key ring creation is reasonably fast.

Data can freely be available on the cloud, and anyone can download it. The only action an unauthorised user can perform is to verify the integrity of the data, because the public key can also be freely available. Since the data is encrypted with the most powerful encryption algorithm, it is practically impossible to decrypt it without the symmetric key. If a key length of 128 bits or longer is used, a brute force attack would not be successful with the current computer technology.

Another action that a malicious user can perform is to modify the data without knowing the content of it, but because the system ensures integrity of data, the authorised users would know that the data has been modified by an intruder whenever they retrieve the data In such a system, whenever the modified data is uploaded, the server would use the corresponding public key to verify the signature, and if the data is not signed with the proper private key, the verification fails, and thus the data is not updated.

Granting Access Permission by Exchanging Keys

In a situation, where two users want to share data with each other, i.e. *A* shares his data with *B*, *A* has to send the key files belonging to the shared data *X*, to *B*. The key files are some or all of the keys belonging to the file *X*, depending on which kind of access permission *A* wants to grant to *B*.

Knowing that the key files must be exchanged in a secure way, one possibility is simply sending them via an encrypted email. Another way is sending them through a secure channel, such as SSL protocol, where communication is done by using an encrypted protocol. And finally, RSA key exchange can be used, where *A* uses *B*'s public key to encrypt the key files, and then sends the encrypted key files to *B*, who can decrypt them by using his private key. In this case the encrypted key files can be sent via email, or it can simply be uploaded to the same place as other files are stored, i.e. the cloud storage. Since it is only *B*, who can decrypt the key files, there is no problem in publishing them. Now if *A* chooses the later solution, i.e. uploading the encrypted key files to the cloud storage, it is necessary to indicate that he has granted access permission to *B*, since there are many other pairs of users that also grant access permissions to each other by publishing their encrypted key files. So, in order for *A* to indicate this, he can use the hash value of *B*'s public key and attach it to the encrypted key files. Then *B* can just perform a search on the hash value of his public key in order to discover the access permission that he has got recently. The reason why *A* uses the hash value of the public key, instead of the very public key, is to avoid directly revealing of the access permission he has given to *B*.

The situation, where many users share data with one user, is fundamentally the same as two users sharing data with each other, and the key exchange process is obviously the same as mentioned above.

Access Permission by Using Key Rings

There is another situation, where the user *A* shares his data with many other users than *B*. In this case it would be more efficient to build a structured system for key exchange in order to reduce the complexities. One of the methods is making use of key rings.

A key ring is a file, which contains key files. Key rings can contain key files, which belong to the stored data and/or other key rings. Key rings must be encrypted, signed and stored to the cloud, and they can be treated the same as other stored files in the cloud. The content of a key ring is just a list of keys. In order to read or update a key ring, it is necessary to have the corresponding key files.

When a user *A* wants to share his data with other users, he puts the corresponding key files on a shared key ring. It can be an existing key ring that has been shared between these users, and *A* has read access to, or a new key ring can be created. If *A* needs to create a new key ring, then it is of course necessary to distribute the key files belonging to the key ring to those users, who should have access to the key ring. The key files belonging to the key rings can be exchanged in one of the ways discussed earlier, namely an encrypted email, an encrypted protocol (SSL), or RSA key exchange mechanism.

If a user wants to give read access to a group of users, but right access to a subset of this group, then he has to make use of two different key rings. Before he distributes each of the two corresponding key files to these two groups, such that every group gets its relevant key files, he obviously has to encrypt the key files with different public keys, Ferguson and Scheier (2003).

Every user can also have his personal key ring, where he puts all his key files, which can belong to the stored data, or other key rings. This personal key ring, as its name indicates, is of course personal, and except its owner, no one has access to it. The method of using key rings is also used in other systems. In Figure 22 an example of key rings is shown. The ellipses represent the key rings, and the squares represent key files. An arrow, which starts from a key ring, indicates that the key set to the object that is pointed to, is contained in the corresponding key ring.

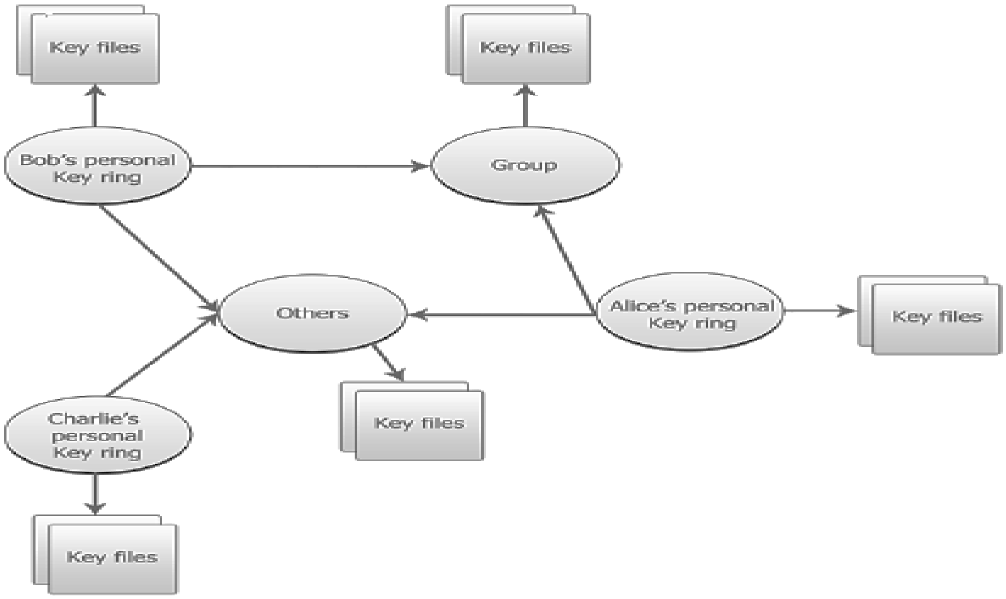


Figure 22: An example of using key rings in a Discretionary Access Control system (DAC)

In Figure 22, Bob and Alice have access to the "Group" key ring, but Charlie does not have access to this key ring. Bob, Alice and Charlie have all access to the "Others" key ring. This type of granting access control to users is used in Discretionary Access Control system (DAC), and this access control mechanism is also used in most UNIX systems. (Kampfeldt, 2003)

Robustness of Cryptographic Access Control

As mentioned earlier, cryptographic access control ensures both confidentiality and integrity of data by using both symmetric and asymmetric cryptography in a hybrid way. In the following sections, there will be a brief discussion on the use both of the encryption methods. The strength and Weaknesses of cryptographic access control with regard to confidentiality and integrity of data will also be examined.

Confidentiality

By comparing the most widely used symmetric and asymmetric algorithms with each other, they are almost equally secure. In terms of key distribution in a large network of users, asymmetric cryptography has less complexity, because it has a pair of keys, and only one of these keys has to be kept secret, while the other is publicly available. So, a user only needs to distribute his private key. In symmetric cryptography, key distribution is more complex, because there is one key that has to be exchanged between all users.

The question is, why not simply use asymmetric encryption for confidentiality of data? To answer this question there should be a mention that asymmetric encryption is very slow, i.e. about 100 to 1000 times slower than symmetric encryption. The data stored by users can often be very large, and thus it would be quite inefficient to use asymmetric encryption. Also, when using asymmetric encryption for digital signature, do not use the actual data. Instead of that generate a hash value of the data, which has a fixed and short length for all types of data. So, it is obviously more reasonable to use symmetric encryption for data confidentiality. There should not be a worry about key distribution issues, because cryptographic access control mechanism is mostly suited for private data stored in the cloud by private users, and thus there will not be a lot of users, who share their data. As a result, there would be rather small network of users, where key distribution would not be such a complex issue.

For ensuring data confidentiality, AES is used. AES is used because its security is very solid. By using a key with a length of 128 bits or more, the algorithm is very powerful and there is no reported issue of any successful attacks against it. The symmetric key is kept secret since it is stored and used on client side. The distribution of the key is also done in a secure way as mentioned in the section "Key Exchange". As a result, there are no serious flaws for data confidentiality.

Integrity

In order to clarify what integrity of data exactly means in cryptographic access control, there will be a mention of the definition of integrity in this context. It is worth specifying that in cryptographic access control mechanism, ensuring the integrity of data does not mean that data must be protected against being lost, corrupted or changed. The loss or corruption of data could happen accidentally or on purpose. Avoiding this problem is mostly the responsibility of the server, and a part of it belongs to data availability.

Assume *A* sends some data to *B*. Integrity of data means to ensure two things:

i. B must know whether or not the data originates from A. For instance, if an intruder pretends to be A and sends some data to B, he would know that it is not coming from A.

ii. B must know whether or not the data has been tampered with by an intruder on its way.

In cryptographic access control mechanism, RSA digital signature scheme for data integrity is used, which is the most widely used scheme. The reason for using this algorithm is that it is a Well-tested algorithm, and thus its security is strong. Through twenty years after invention of RSA algorithm, a lot of attacks, both on the implementation of RSA and on the actual algorithm, have been developed to find Weaknesses of RSA. By following the protection guidelines, none of these attacks would be successful. On the other hand, some of these attacks are not applicable in cryptographic access control. For instance, timing attack (cf. § 2.1.3.2) is not applicable, because the whole process of signing the data is performed on a local client. The common modulus attack (cf. § 2.1.3.2) is not applicable either, because every user generates the pair of keys on their own machines, and thus the value of *n* in RSA algorithm would be different for every user. As a result, there are not any serious flaws for data integrity. To sum up, the attacks on the actual data without the presence of corresponding keys would not be successful. If it is assumed that the keys are kept secret and are not accessible in any way, then the confidentiality and integrity of data are guaranteed.

Implication on future Research

In this security solution, cryptography is used to ensure confidentiality and integrity of the stored data. The main quality in this solution is that the security operations are performed at the client side, and thus the users do not need to trust the cloud servers. The only elements that make it possible to access the stored data are the corresponding keys, and thus file sharing between users can only happen by exchanging keys.

CONCLUSIONS

In this paper, a solution for data confidentiality and integrity in cloud storage systems is examined. The available solutions in the market are studied, whereupon the possibilities for a solution based on cryptography is analyzed, and as a result the cryptographic access control mechanism is proposed. This exploration shows that there is not any standard or a common agreement regarding the security solutions in the cloud. Various solutions are used by different providers, usually in a hybrid way. For instance, Amazon uses ACLs (Access Control Lists) as an access control mechanism, and SSL channel for data transfer, and besides they also use cryptography for data confidentiality.

Suggestions for future research

Future research could be geared towards making the proposed system a crossed platform based since the current system is solely windows based even though java language with NetBean was used. The proposed Algorithm can use more complex operation to increase security level. Various algorithms can be applied randomly instead of using constant algorithm; it can use random order algorithm for compression and encryption.

REFERENCES

Aamer Nadeem et al. (2005). *A Performance Comparison of Data Encryption Algorithms,*. IEEE. Pg 57-80

About Dropbox. (n.d.). Retrieved from https://www.dropbox.com/about. (accessed on 20-06-2018)

Ajay K., S. M. (January, 2012). Comparison of Various Encryption Algorithms and Techniques for Secured Data Communication in Multinode Network. *International Journal of Engineering and Technology Volume 2 No. 1, ISSN: 2049-3444 © 2011 – IJET Publications UK.*, 87-92.

Ankita P. B., L. S. (April - 2013). A Comparative Literature Survey On Various Image Encryption Standards. *International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 4, ISSN: 2278-0181*, 1444-1450.

Charles P. Pfleeger. (2006). *Security in Computing, Fourth Edition,*. Prentice Hall: Pfleeger Consulting Group, Shari Lawrence Pfleeger - RAND Corporation. Retrieved from http://www.infoq.com/articles/infinispan-gridfs: http://www.infoq.com/articles/infinispan-gridfs (accessed on 17-11-2018)

Christof Paar and Jan Pelzl. (Springer 2010). *Understanding Cryptography:*. A Textbook for Students and Practitioners,. pg 102-156

Dropbox/8301-31921_3-20072755-28. (n.d.). *http://news.cnet.com/8301-31921_3-20072755-281/dropbox-confirms-security-glitch-no-password-required/*. Retrieved from http://news.cnet.com: http://news.cnet.com/8301-31921_3-20072755-281/dropbox-confirms-security-glitch-no-password-required/ (acceseed on 12-04-2018)

Google. (2009, 02 25). *Google Gmail outage on data centre collapse,*. Retrieved from google_gmail_data_centre_fail/: http://www.theregister.co.uk/ accessed on 13-12-2014

Hellman, W. D. (June, 1977). "Exhaustive Cryptanalysis of the NBS Data Encryption Standard" Computer. 74-78.

Jensen., A. H. (June 2-3, 2003.). Cryptographic Access Control in a Distributed File System. In *In SACMAT'03,* (pp. pages 158-165,). Italy: Como.

Kakkar A., S. M. (January, 2012). Comparison of Various Encryption Algorithms and Techniques for Secured Data Communication in Multinode Network . *International Journal of Engineering and Technology (IJET) – Volume 2 No. 1* , 87-92 .

Kampfeldt, S. H. (2003). Kryptografisk adgangskontrol i peer-to-peer netværk. *Msc Thesis, IMM DTU,*.

Khovratovich, A. B. (2009/317). *Related-key Cryptanalysis of the Full AES-192 and AES-256,*. University of Luxembourg,: ePrint Archive:.

Mandar M. K., P. B. (March 2013). Encryption Algorithm Addressing GSM Security Issues-A Review. *International Journal of Latest Trends in Engineering and Technology (IJLTET), Vol. 2 Issue 2 , ISSN: 2278-621X*, 268-273.

Nikos Virvilis, S. D. (Heidelberg ©2011). Secure Cloud Storage:Available Infrastructures and Architectures Review and Evaluation,. *Springer-Verlag Berlin,* (pp. TrustBus'11Proceedings of the8th international conference on Trust,). TrustBus'11.

NIST. (2001 Edition,). *, Recommendation for Block Cipher Modes of Operation,*. NIST Special Publication 800-38A,.

Pelzl, C. P. (Springer 2010). *Understanding Cryptography.*

Pfleeger, C. P. (2006). *Security in Computing, Fourth Edition,*. Prentice Hall,: Pfleeger Consulting Group, Shari Lawrence Pfleeger - RAND Corporation,.

Ruohonen, K. (2010). *Finnish lecture notes for the TUT course "Matemaattinen kryptologia".* (Translation by Jussi Kangas and Paul Coughlan).

Salesforce.com. (n.d.). *http://searchcrm.techtarget.com/news/1281107/Salesforce-com-customers-hit-with-phishing.* Retrieved from http://searchcrm.techtarget.com: http://searchcrm.techtarget.com/news/1281107/Salesforce-com-customers-hit-with-phishing-attack: (accessed on 20-09-2018)

Schneier, B. (1996). *Applied Cryptography. Protocols, Algorithms, and Source Code in C. 2nd edition.* New York: John Wiley & Sons.

Shah K. R., B. G. (March, 2012). New Approach of Data Encryption Standard. *International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Volume-2, Issue-1,*, 322-325.

Sharma, H. A. (2010). Implemenentation and analysis various symmetric cryptosystems. *Indian Journal of Science snd Technology Vol. 3 No. 12 ISSN:0974-6846*, 1173-1176.

Standaert, F. R. (2003). *Applications: A methodology to implement block ciphers in reconfigurable hardware and its application to fast and compact AES RIJNDAEL".*

Top five cloud computing security issues. (n.d.). *http://www.computerweekly.com/news/2240089111/.* Retrieved from http://www.computerweekly.com/: Top-five-cloud-computing-security-issues

Vikendra S., a. S. ((2013)). ANALYSING SPACE COMPLEXITY OF VARIOUS ENCRYPTION ALGORITHMS. *International Journal of Computer Engineering and Technology (IJCET), ISSN 0976-6367(Print), ISSN 0976 – 6375(Online) Volume 4, Issue 1,*, 414-419.

What are the system requirements to run Dropbox?,. (n.d.). Retrieved from , https://www.dropbox.com/help/3: https://www.dropbox.com/help/3

Yogesh K., R. M. (Oct 2011). Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures. *IJCSMS International Journal of Computer Science and Management Studies, Vol. 11, Issue 03, ISSN (Online): 2231-5268*, 60-63.