future internet

MDPI

Article

Security Enhancement for Data Migration in the Cloud

Jean Raphael Ngnie Sighom *, Pin Zhang and Lin You

School of Communication Engineering, Hangzhou Dianzi University, Xiasha Higher Education Zone, Hangzhou 310018, China; zhangpin@hdu.edu.cn (P.Z.); mryoulin@gmail.com (L.Y.)
* Correspondence: raphaelngnie@yahoo.fr; Tel.: +86-186-5717-1562

Academic Editor: Luis Javier Garcia Villalba
Received: 30 May 2017; Accepted: 16 June 2017; Published: 22 June 2017

Abstract: In today's society, cloud computing has significantly impacted nearly every section of our lives and business structures. Cloud computing is, without any doubt, one of the strategic directions for many companies and the most dominating infrastructure for enterprises as long as end users. Instead of buying IT equipment (hardware and/or software) and managing it themselves, many organizations today prefer to buy services from IT service providers. The number of service providers increase dramatically and the cloud is becoming the tools of choice for more cloud storage services. However, as more personal information and data are moved to the cloud, into social media sites, DropBox, Baidu WangPan, etc., data security and privacy issues are questioned. Daily, academia and industry seek to find an efficient way to secure data migration in the cloud. Various solution approaches and encryption techniques have been implemented. In this work, we will discuss some of these approaches and evaluate the popular ones in order to find the elements that affect system performance. Finally, we will propose a model that enhances data security and privacy by combining Advanced Encryption Standard-256, Information Dispersal Algorithms and Secure Hash Algorithm-512. Our protocol achieves provable security assessments and fast execution times for medium thresholds.

Keywords: Advanced Encryption Standard (AES-256); Information Dispersal Algorithms (IDAs); Secure Hash Algorithm (SHA-512); Cauchy Reed Solomon (C-RS)

1. Introduction

Over the past several decades, our IT society has been overwhelmed by a new buzzword of "going Cloud". This new buzzword refers to a business moving toward cloud computing. Cloud computing is commonly used to represent any work done on a computer, mobile or any device, where the data and possibly the application you are using do not reside on your device but rather on an unspecified device elsewhere on the Internet. The basic premise of cloud computing is that consumers (individuals, industry, government, academia and so on) pay for IT services from cloud service providers (CSP). Services offered in cloud computing are generally based on three standard models (Infrastructure-, Platform-, and Software as a Service) defined by the National Institute of Standards and Technology (NIST) [1]. There are also a ton of services on the internet that consumers can take advantage of. As more cloud-based services available to users, their oceans of data are outsourced in the cloud as well. The cloud becomes, then, the tools of choice for more data storage services.

1.1. Cloud Data Storage versus Traditional Data Storage

Compared to traditional data storage as shown in Figure 1, cloud data storage offers numerous advantages:

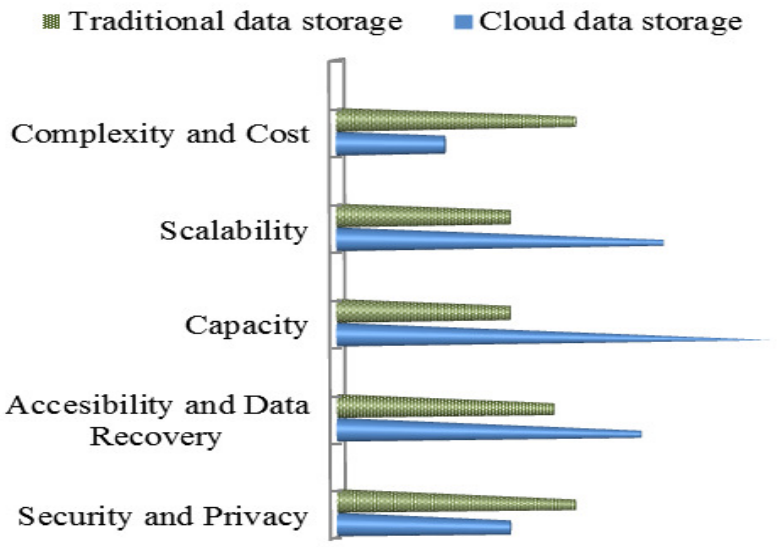

Figure 1. Traditional storage vs. cloud storage.

1.1.1. Lower Complexity and Costs

Almost six out of 10 providers say that cost reduction is the main goal for customers to use cloud services [2]. Consumers do not need to buy and configure new equipment. Cloud storage service allows them to get their application started immediately, and the providers charge the users based on the usage of resources (Pay As You Go). The service usually costs a fraction of what it would cost to implement an on-site solution.

1.1.2. Scalability and Capacity

As the consumer requires more capacity, the service provider can make more scalability, much simpler than if you had to add the equipment on your own premise. Therefore, you can scale your storage capacity regardless of the amount of space that you need.

1.1.3. Archival and Disaster Recovery Purposes

Cloud storage provider can help the organization enhance security from Internet services, by preventing loss due to fire, theft, or disaster. Therefore, companies who archive their information in the cloud do not have to worry too much about natural disasters.

1.1.4. Greater Accessibility and Reliability

Stored files can be accessed from anywhere via Internet connection. However, there are many other benefits to adopting cloud storage services, and there are also some factors affecting cloud adoption such as Usability, Bandwidth, Data Security and Privacy, and so on. Indeed, the two most significant barriers to cloud adoption are Security and Privacy. Except for these two barriers, the others are beyond the scope of our research.

1.2. Cloud Storage Security and Privacy Threats

For all the advantages that provide cloud storage, we cannot ignore the potential risks that face our data when storing them in the cloud environment. Most of the time, consumers move information into the cloud without any consideration of security. Once their data is outsourced to the cloud service providers, they have to trust the CSP, which means that data is out of control and the service provider can access the data in the cloud at any time. It could accidentally or deliberately alter, even delete, information [3].

Several challenges need to be addressed with respect to security and privacy in cloud storage-ensure authentication, authorization and access control mechanisms; prevent data loss or leakage; secure cloud credential; guaranteeing control and manage the cryptography key; and finally secure data in transit. This has been supported by the NASUNI (Boston, MA, USA) vendor in their press releases of top five security challenges of cloud storage [4].

There are some overwhelming questions that are still causing consumers to delay their move to the cloud: can we trust CSP to keep our data safe? How is sensitive data be secured in the Cloud?

Paper Organization: Section 2 gives preliminary definitions and formally introduces the notion of the Information Dispersal Algorithm. In Section 3 we analyze, discuss, and evaluate some existing solutions and prior works. Section 4 describes our proposed methodology. Finally, Section 5 presents the performance analysis and evaluation of our proposed approach.

2. Background Preliminaries

2.1. Cryptographic Algorithms

This is the study of techniques for ensuring the secrecy and/or authenticity of information [5]. The three main cryptographic algorithms are: Symmetric-Key Algorithm, Asymmetric-Key Algorithm and Hashing Algorithms.

2.1.1. Symmetric-Key Algorithm Also Known as Secret Key Encryption

A single key is used for both encryption and decryption. The key, in practice, represents a shared secret between two or more communicating parties for secure communication. A few of the well-known symmetric encryption algorithms include Data Encryption Standard (DES), AES, Blowfish and Skipjack.

2.1.2. Asymmetric-Key Algorithm or Public-Key Encryption

The asymmetric-key algorithm or public-key encryption utilizes a pair of keys—public key and private key. The public key can be revealed, but, to protect the data, the private key must be concealed. Additionally, encryption and decryption of the data must be done by the associated private and public keys. For instance, data encrypted by the private key must be decrypted by the public key and vice versa. Some of the common examples of this algorithm are RSA (Rivest-Shamir-Adleman), Diffie–Hellman key exchange, and elliptic curve techniques.

2.1.3. Hashing Algorithms, Also Called One-Way Encryption

Hashing algorithms, also called one-way encryption, are algorithms that, in some sense, use no key. It is a vital information security tool and is used to authenticate messages, digital signatures and documents. A hash function accepts a variable-length block of data as input and produces a fixed-length hash value called message digest. Having a message digest, it is impossible to recover or find the original string. Some examples of hashing algorithms are: MD5, SHA-1 SHA-2, and SHA-3.

The primary purpose of encryption is to ensure the confidentiality of the data stored on a specific device or transmitted via the internet. While hash function ensures data integrity, any change in the original message, however small, must cause a change in the digest, and, for any given file and digest, it must be infeasible for a forger to create a different file with the same digest [3].

2.2. Data Masking or Obfuscation

In today's information era, data is a key asset for any organization. It is also the currency of the 21st century. Data masking, as the name suggests, is a process that disguises or conceals the sensitive data in the database by implementing a particular mathematical function or using programming techniques. The point is to provide data that looks and acts like the original data, but which lacks

sensitivity and does not pose a risk of exposure. Therefore, masking an e-mail address in an online application [6] produces the corresponding results:

the Unmasked e-mail address is: username@qq.com;
masking result: email@HOSTNAME.REDACTED.

Just like encryption, apply masking or obfuscation to a data will produce a suitable way of addressing data confidentiality concerns. The efficiency of the data masking technique used has to be such that the original data cannot be reconstructed from the modified data unless the masking technique is known. However, data masking evolves a variety of techniques—Substitution, Shuffling, Number and Date Variance, Encryption, Nulling out or Deletion, Masking outs and so forth. Deciding to apply data masking requires that the user specifies which data masking features will be used on information since it has many capabilities, and not all are appropriated to maintain valid business contextual information [7].

2.3. Information Dispersal Algorithm

An IDA is the art of splitting a file and data packet into pieces at a bit level so that they are unrecognizable as they sit in storage arrays or traverse the network. Data can be reassembled at the receiving end user or device using the right key.

First considered by Rabin [8], his schemes are intended to break a file F of length $L = |F|$ into n pieces F_i, and each length $|F_i| = L/m, 1 \leq i \leq n$, so that every m pieces suffice for reconstructing F. IDA can address better issues of confidentiality, integrity and availability of data.

Basically, an information dispersal scheme can be implemented in a variety of ways, all corresponding to the notion of erasure codes in the theory of error correcting codes. An IDA that adopts erasure code has strong confidentiality, whereas IDA with an arbitrary non-systematic erasure code may fall into weak confidentiality [9]. IDA has numerous applications to secure and reliable storage of information in computer networks and even on single disks. Nowadays, vendors such as CleverSafe, Tahoe-LAFS, among others, have their technology relying on IDA.

3. Existing Solutions and Prior Works

Protect data outsourced in the cloud has always been the biggest challenge for IT providers and academia to promote cloud adoption. Pick up any IT magazine or browse almost any social media site or IT blog, and you will be sure to see talk about data security and privacy. More and more people are complaining about personal data breach, and enterprises as well recognize that it is critical to move sensitive data in the cloud without knowing the appropriate security control put in place.

These days, numerous vendors and cloud providers are leveraging cloud-based storage models to deliver their security solutions known as Security as a Service (SecaaS). It is likely for any other cloud services that the SecaaS model embraces all of the characteristics of cloud computing and it provides to users less or no control over services and tasks. The Cloud Security Alliance (CSA) has identified some services offered by SecaaS [10]—among them is Encryption as a Service.

According to some providers such as Amazon Web Services (AWS) and Dropbox, Encryption as a service is the most obvious way to keep data secure in the Cloud environment [11]. For instance, AWS offers a flexible and scalable layer of security for data stored in the cloud by providing several options for encrypting data at rest-ranging from completely automated AWS encryption solutions to manual and client-side options; however, in reality, they don't provide on-premises key management controls and data access policy option for EC2 and EBS data [12].

SecaaS offerings can take many forms causing market confusion and complicating the selection process. Customers are increasingly faced with evaluating SecaaS solutions, which do not run on premises, and they need a better understanding of these offerings to assess the type of security issues they address and also to evaluate the security risks and the shared responsibility for the security of systems for which they are accountable.

Even though CSPs claim that customer data is well safe and secure, we know, from reports [13] of cloud leaks on celebrities and companies, that there is no complete guarantee of safety when storing sensitive data in the cloud environment unless consumers approach cloud in the right way, with the correct checks and balances to ensure that all security and risk management measures are covered.

In addition to SecaaS, academia and industry have developed a lot of research works and practical approaches to secure cloud data storage.

Many recent proposed models are based on cryptographic algorithm [14–17] such as DES, AES and Elliptic curve cryptography (ECC). Therefore, to ensure the secrecy of data before migrating in the cloud, some methods relied upon data encryption by crypting data using AES-128, 192, 256 bits, depending on the file size and data format [17]. 128 and 192 bits key encryption are suggested to users to encrypt, respectively, large and medium data files. However, encryption with 256 bits key will be done on small data files, but based on the data format rather than its size.

Another approach is to combine Encryption and Obfuscation [18,19]. Encryption is done on alphabets and alphanumeric type of data, and obfuscation only to the numeric type of data [18]. Specifically, encryption helps a user to provide secrecy to its data when it is transferred in the network, and data obfuscation helps the service provider to secure the user's data at rest [19]. This model provides an improvement of data confidentiality.

Secure personal cloud storage system that affords the plug-n-play convenience of portable storage device (PSD) is highlighted [20]. Authors proved, the accessibility and scalability of cloud storage, using the information dispersal algorithm (IDA) to address the dual requirements of data confidentiality and availability. Their implementation allows for slices files to be spread across multiple CSPs. Therefore, the adversary will need to compromise at least two over three cloud providers to be able to reconstruct the data. Similarly, two CSPs have to collude to mount any effective attack.

Zhang X. and Wang H. proposed a system architecture adopting IDAs for securing off-site data storage [21]. The key component of this system is a proxy server. For the process of writing a file to cloud storage, the user copies a file to a desired folder on the network drive. This file will be cached on the proxy server; almost at the same time, the proxy server will generate a special random matrix to transform the file into multiple slices. In order to further enhance the confidentiality of information, the data fragments will be encrypted by the proxy server before they leave the trusted intranet. The resulting data slices will be stored on the online cloud drives provided by Amazon, Dropbox, or Rackspace via the protocol adapter.

Indubitably, to ensure the secrecy of data, a vast majority of secured cloud storage solutions available today relies upon symmetric and/or asymmetric data encryption. However, encryption is good but not enough since it is only computationally secure; an advancement in cryptanalysis may render what is considered a secure encryption today ineffective tomorrow [20]. Moreover, the promised potential of quantum computing may further render existing cryptographic algorithms obsolete [22].

4. Proposed Model

There is a big wall when adopting the cloud, which is security of data. The cloud has many challenges and issues, but, among that, security is the main concern. Mostly, cryptography techniques are preferred for securing data and most of the techniques are outdated [23]. Therefore, to overcome the Security and Privacy issues in cloud storage, we proposed a model that ensures data confidentiality, integrity, availability and data leakage as well.

Our proposed approach consists of encoding or decoding data on the premise of adopting the Advanced Encryption Standard with a 256-bits key (AES-256), IDA with C-RS and then the SHA-512 hashing algorithm as the main components of the process. Table 1 below defined some symbols and parameters used in this paper.

Table 1. Notation and parameter description.

Notations	Description
F	Original data
F'	Encrypted data
F''	Slices set
F'''	Encoded set
f_i	Slice data file
ξ, Δ	Encryption, Decryption
κ	Encryption key
$f_i' = H(f_i)$	Hash value using SHA-512
$\mid \mid$	Concatenation
$S = \lvert F' \rvert$	Encrypted file length
n	total number of slices
m	number of symbols/bytes required to reconstruct F'
$\omega = \lvert f_i \rvert$	Slice data file length

4.1. Overview of the Encoding Process

The encoding process consists of three levels of security: encryption with 256-bit key length, IDA and Hashing algorithm.

During the encoding process in Figure 2, the user original data or file F is firstly encrypted using the AES-256 algorithm. The encryption key κ is generated randomly, yet managed and protected by the consumer or the IT manager [24]. Secondly, the preceding encrypted data file $F' = \xi(F, \kappa)$ is broken into n separated slices file in such a way that knowledge of at least any m of these n slices can be used to reconstruct the encrypted data file F'.

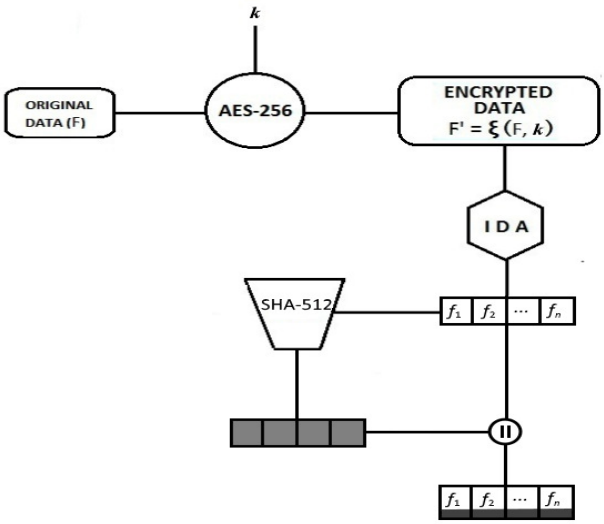

Figure 2. Encoding steps.

Assume $F' = (Y_1, \cdots, Y_S)$ be a file of size S (measured in some scale i.e., strings symbols or bytes). The symbols Y_i is interpreted as an element in some finite field $\mathbb{F}$ [25]. The file F' can be split into blocks of m symbols or bytes. Thus, $F' = (Y_1, \cdots, Y_m), (Y_{m+1}, \cdots, Y_{2m}), \cdots, (Y_{S-m+1}, \cdots, Y_S)$.

We lay out the file F' as an $m \times \omega$ matrix, where ω is the size of each data slice:

$$\Omega_{m\times\omega} = \begin{pmatrix} Y_1 & Y_{m+1} & \cdots & Y_{(\omega-1)\cdot m+1} \\ Y_2 & Y_{m+2} & \cdots & Y_{(\omega-1)\cdot m+2} \\ \vdots & \vdots & \ddots & \vdots \\ Y_m & Y_{2m} & \cdots & Y_{\omega\cdot m} \end{pmatrix}; \quad \omega = \left(\frac{S}{m}\right). \tag{1}$$

Hence, $Y_i = Y_{(i-1)m+1}, Y_{(i-1)m+2}, \cdots, Y_{im}$ are the i^{th} blocks of m symbols. Consider G to be the $n \times m$ Cauchy matrix used to transform the original matrix Ω into n slices:

$$G_{n\times m} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}.$$

Each subset of m rows is a linearly independent vector [26]. To disperse, we multiply G and Ω, which yields an $n \times \omega$ matrix δ:

$$G \cdot \Omega = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \cdot \begin{pmatrix} Y_1 & Y_{m+1} & \cdots & Y_{(\omega-1)\cdot m+1} \\ Y_2 & Y_{m+2} & \cdots & Y_{(\omega-1)\cdot m+2} \\ \vdots & \vdots & \ddots & \vdots \\ Y_m & Y_{2m} & \cdots & Y_{\omega\cdot m} \end{pmatrix}$$
$$= \begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1\omega} \\ f_{21} & f_{22} & \cdots & f_{2\omega} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1} & f_{n2} & \cdots & f_{n\omega} \end{pmatrix} = \delta_{n\times\omega}$$

where $f_{ij} = \sum_{k=1}^{m} a_{ik} \cdot Y_{(j-1)m+k}; \quad 1 \leqslant i \leqslant n; \quad 1 \leqslant j \leqslant \omega.$

Given Ω, a dispersal algorithm can produce a matrix δ consisting of n rows and ω columns. Each row of δ corresponds to a specific file slice: $f_i = (f_{i1}, f_{i2}, \cdots, f_{i\omega})$.

Therefore, applying IDA to F' will produce F'' consisting of n slices. Any m out of n can be used for reconstruction:

$$F'' = (f_{11}, \cdots, f_{1\omega}), \cdots, (f_{i1}, \cdots, f_{i\omega}), \cdots, (f_{n1}, \cdots, f_{n\omega}). \tag{2}$$

Finally, we compute the hash value of each slices f_i using SHA-512 and concatenates its coresponding hashing value f_i' to produce $(f_i||f_i')$. This results in

$$F''' = f_1||f_1', f_2||f_2', \cdots, f_n||f_n'. \tag{3}$$

The slices data file in F''' can now be uploaded in the cloud through a secure connection. Applying our protocol algorithm to consumer data will overcome the problem of data leakage and corruption. Furthermore, the user can now store the dispersed slices data resulting from Equation (3) in at least three different CSPs [20] so that, even if one CSP is hacked or one server goes out of service, the user can recover the files from other CSPs. Table 2 below shows the encoding algorithm.

Table 2. Encoding algorithm pseudo code.

Procedure
Input: Original data F; key κ, Threshold (m, n)
Output: F'''
(Note that AES_{256} is the encryption algorithm, SHA_{512} is the hashing algorithm)
compute the encrypted version of the original data file F using AES_{256} under κ
F':= AES_{256} with F and κ
Let IDA_{CRS} be the slicing algorithm that return a vector F'' under the threshold (m,n)
F'':= IDA_{CRS} with F', m and n
set indexCount **to** 1
set F''' := new initial list
for each f_i in F'' **do**
set f_i' := **call** SHA_{512} with f_i
concatenate the data f_i with the corresponding hash value f_i'
set $concat_{(f_i,f_i')} := f_i + f_i'$
Append to F''' the concatenated data $concat_{(f_i,f_i')}$
$F'''[indexCount] := concat(f_i, f_i')$
increment indexCount
return: F'''
end procedure.

4.2. Overview of the Decoding Process

During the decoding process, the verification step is first observed in Figure 3a. This step consists of checking the integrity of the file and deconcatenating the m correct slices data required for reconstructing F'. In the case of slice corruption, an authentication message will be displayed, and the corresponding slice data will be replaced by another one.

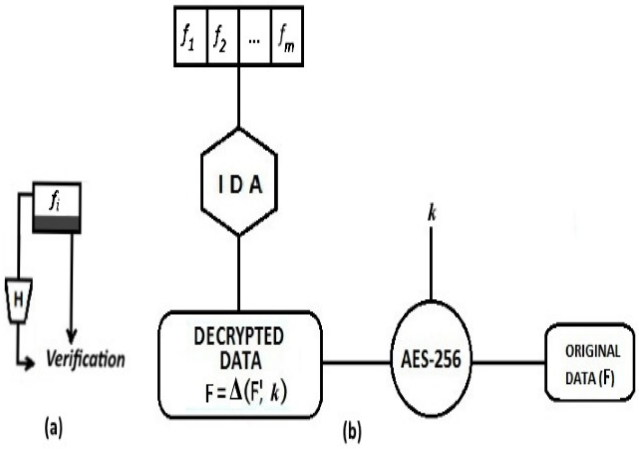

Figure 3. Decoding steps: (**a**) Verification step; (**b**) reconstruction step.

After the verification step done, we apply IDA to the m resulting slices as shown in Figure 3b, to reconstruct the encrypted data F' and finally recovert the original data file F having the encryption key κ with AES_{256} as detailed in Figure 3. For convenience, we assume no error due to malice or accidents existing in the first m slices use for recombination. Consider δ' an $m \times \omega$ submatrix of δ and G' be an $m \times m$ principal submatrix of G. According to Rabin's IDA [8], G is a Cauchy matrix in which any square submatrix is nonsingular. Thus, G' is invertible. Let G'^{-1} be the inverse of G', and, computing $G'^{-1} \cdot \delta'$, results in Ω.

$$G'^{-1} \cdot \delta' = \begin{pmatrix} \partial_{11} & \partial_{12} & \cdots & \partial_{1m} \\ \partial_{21} & \partial_{22} & \cdots & \partial_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \partial_{m1} & \partial_{m2} & \cdots & \partial_{mm} \end{pmatrix} \cdot \begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1\omega} \\ f_{21} & f_{22} & \cdots & f_{2\omega} \\ \vdots & \vdots & \ddots & \vdots \\ f_{m1} & f_{m2} & \cdots & f_{m\omega} \end{pmatrix}$$
$$= \begin{pmatrix} \Upsilon_1 & \Upsilon_{m+1} & \cdots & \Upsilon_{(\omega-1)\cdot m+1} \\ \Upsilon_2 & \Upsilon_{m+2} & \cdots & \Upsilon_{(\omega-1)\cdot m+2} \\ \vdots & \vdots & \ddots & \vdots \\ \Upsilon_m & \Upsilon_{2m} & \cdots & \Upsilon_{\omega \cdot m} \end{pmatrix} = \Omega,$$

where $\Upsilon_j = \sum_{t=1}^{m} \partial_{it} \cdot f_{tk}; 1 \leqslant j \leqslant S; i \equiv j \bmod [m]; k = \left[\frac{j}{m}\right]$.

Hence, F' is reconstructed having the first m slices data as shown in Equation (1).

Finally, using AES-256, we can recompute F with the same secret key κ used during the encryption step in Figure 2, $F = \Delta(F', \kappa)$. Because the secret value is kept by the user or IT manager, an opponent cannot modify an intercepted piece of data and cannot generate a false message [5]. Table 3 shows the decoding algorithm.

Table 3. Decoding algorithm pseudo code.

Procedure
Input: vector H; key κ,value m
Output: Original data F
(Note that AES_{256} is the encryption algorithm, SHA_{512} is the hashing algorithm)
let integrityCheck function checks the integrity of a particular data $concat_{(f_i,f_i')}$
set indexCount **to** 1
set F' := new initial list
for each $concat_{(f_i,f_i')}$ in H **do**
if integrityCheck with $concat_{(f_i,f_i')}$ **then**
deconcatenate f_i from $concat_{(f_i,f_i')}$
compute f_i := $concat_{(f_i,f_i')}$ - f_i'
else
compute $H[indexCount]$
Append to $F''[indexCount]$ the deconcatenated data f_i
set F'' := f_i
Let IDA_{CRS} be the slicing algorithm that return a vector F' under m
Reconstruct F' using IDA_{CRS}
$F'[indexCount]$:= IDA_{CRS} with F', m
Compute the decrypted version of F using AES_{256} under κ
F:= AES_{256} with F' and κ
return: F
end procedure

5. Experimental And Result Evaluations

To assess the actual performance, we analyzed the performance of our algorithm by measuring the computation time of the encoding and decoding process for a data file, depending on the size (51 KB to 347,778 KB) and a variability of threshold (m, n); $1 \leq m \leq n$ and $n \leq 256$.

We carry out these experiments on a Windows 7 Operating System (Redmond, WA, USA) with 8-core Intel Xeon E5-1620 (Santa Clara, CA, USA) at 3.50 GHZ with 32 GB of memory.

In Figure 4, we have shown the computation time of the encoding operation. The encoding time for our algorithm mainly depends on the size of the data and the value of the (m, n) threshold. We obtained the computation time by summing the encryption time of x KB data plus the time to execute IDA with C-RS, and the time to hash and concatenate each slices file. The maximum encoding

time is 14.15 s for a large data file equal to 347,778 KB with $(200, 254)$ threshold. We observed for different (m, n) configurations and data less than 347,778 KB that our encoding algorithm yields the best performance, since the average encoding time is 1.966 s.

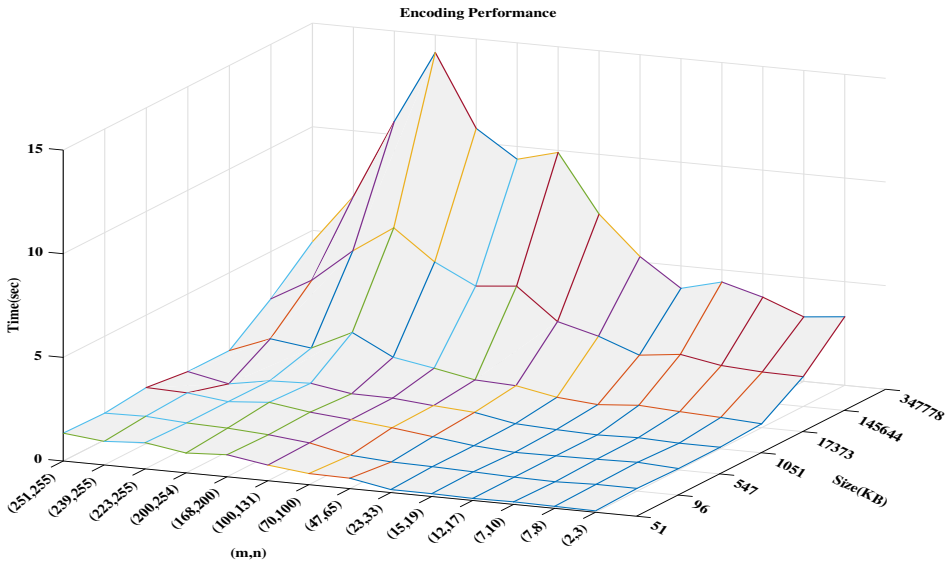

Figure 4. Encoding results.

In Figure 5, we have shown the running time of the decoding operation. As we have described in Section 5, the decoding time for x KB data file is equal to the sum of the different steps followed during the process. The maximum computation time is 24.30 s for 347,778 KB file with (2,3) and the average decoding time in this experiment is 2.907 s. The decoding process provides the highest performance for small and medium data size. However, the time required to decode x KB data depends primarily on the inner verification time and secondly on the reconstruction time. For large data size, these two processes become a bit computationally costly, but better than the preceding works. Depending on the value of (m, n), when the threshold is small, the verification time increases and the reconstruction time decreases. In addition, when the threshold is large enough, the verification time decreases dramatically and the reconstruction time increases. Therefore, we recommend, for the best performance, to use a medium threshold. This will considerably reduce the decoding computation time.

In Table 4 below, we displayed the average execution time for each inner step in the encoding and decoding processes. It is quite obvious that the encoding time is principally estimated by the slicing execution time, since it requires more parameters and computations than the other inner processes.

Table 4. Average encoding/decoding time.

Process	Duration
Encryption	0.367
Slicing-IDA	1.125
Hashing and Concatenation	0.474
Verification	1.453
Reconstruction-IDA	1.157
Decryption	0.297

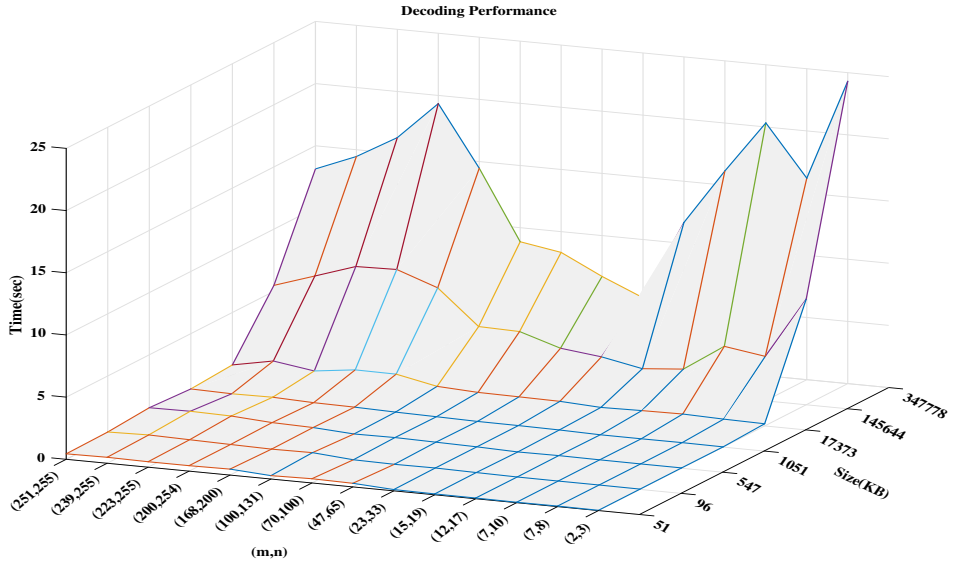

Figure 5. Decoding results.

Encryption is everywhere; most of the products gathered today support a cryptographic algorithm and particularly AES. Even those that support other algorithms tend to recommend using AES [27]. However, most of the cryptographic algorithms (AES, RSA, etc.) are not secure against quantum computing [28]. In the following lines, we are especially looking at products that encrypt files, not at whole-disk solutions like VeraCrypt.

Boxcrypor: is a file-encryption software designed and created specifically for cloud use, with support for all major cloud-storage providers. The user has the ability to encrypt, decrypt, share, add and remove files from cloud storage directly through his/her end device. Furthermore, there is a possibility to protect files by encrypting them locally before posting to a cloud-storage provider, ensuring file security and file-data privacy throughout the process. Boxcryptor uses AES-256 and RSA encryption algorithms. It ensures data accessibility, confidentiality and privacy protection.

SecureDoc CloudSync: designed by WinMagic, is an enterprise endpoint encryption solution that encrypts data at rest on endpoint devices before they are stored in cloud, providing an added layer of security to that offered by the storage service provider. SecureDoc CloudSync uses Advanced Encryption Standard–New Instructions (AES-NI) 256-bit encryption. It provides data confidentiality and data privacy but does not prevent eavesdropping.

In regard to these two product solutions and their deployement, it is quiet obvious that they provide faster execution time than the proposed methodology since both are relying upon encryption, which corresponds to the first layer of our algorithm (AES-256). However, their use ensures data confidentiality and privacy but does not guaranty data integrity and availability. However, our implementation enhances data CIA (Confidentiality-Integrity-Availability) [5]. Confidentiality is achieved by combining AES-256 and IDA, integrity by concatenating each particular slice file with its corresponding SHA-512 hashcode, and, finally, data availability (Section 2.3) is accomplished by dispersing (IDA) the resulting data slices, as we suggested earlier in Section 4.1.

6. Conclusions

Cloud computing is a multi-tenant environment, where resources are shared. Threats can happen from anywhere; inside or outside the shared environment. Deciding whether to migrate sensitive data or keeping it on premise is one of the most important decisions faced by personal users, as well as small and medium-sized enterprises.

We have described throughout this paper a number of key risks associated with cloud data storage and examined the existing ways of mitigating data security and privacy threats. Our proposed model based on the combination of AES-256, IDAs and SHA-512 consists of encoding and decoding data on premise. The encoding operation includes: AES-256 Encryption to ensure data confidentiality, IDA with Cauchy Reed–Solomon code to break the encrypted data into n slices such that we can recover from m and then SHA-512 Hashing algorithm for signature. However, the decoding operation includes a verification process to check data slice integrity, IDAs to reconstruct the encrypted data from m slices, and, finally, the decryption process to recover the original data. Compared to other approaches to secure data privacy, our algorithm achieves far a greater degree of security and also better performance for small and large data files. Our implementation allows slices data to be spread across multiple CSPs. In addition, for higher performance, we suggest choosing a medium threshold value of (m, n) for encoding and decoding operation.

Acknowledgments: This work was supported by the Key Program of the Natural Science Foundation of Zhejiang province, China (No. LZ17F020002).

Author Contributions: Jean Raphael Ngnie Sighom conceived and designed the algorithm, performed and implemented the experiments, and wrote the paper. Both Pin Zhang and Lin You have fully supervised the work and approved the paper for submission.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. NIST, Cloud Computing Program–29 July 2016: Cloud Computing. Available online: https://www.nist.gov/programs-projects/cloud-computing (accessed on 13 January 2017).
2. Wolfgang, G. Cost Reduction Remains Chief Reason to Adopt Cloud, Confusion Still Apparent. Available online: http://www.tomsitpro.com/articles/cloud_survey-kpmg-tech_adoption-provider-it_security,1-803.html (accessed on 7 January 2017).
3. Cloud Computing. Available online: https://en.wikipedia.org/wiki/Cloud_computing#Security_and_privacy (accessed on 8 February 2017).
4. Press Releases. Cloud Storage Security Challenges. Available online: https://www.nasuni.com/news/26-top_5_security_challenges_of_cloud_storage/ (accessed on 5 April 2017).
5. Stallings, W. *Cryptography and Network Security Principles and Practices*, 6th ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2005.
6. Mask Sensitive Data. Available online: https://dataapps.io/redact.html (accessed on 12 January 2017).
7. Sarada, G.; Abitha, N.; Manikandan, G.; Sairam, N. A few new approaches for data masking. In Proceedings of the 2015 International Conference on Circuit, Power and Computing Technologies, Nagercoil, India, 19–20 March 2015.
8. Rabin, M.O. Efficient dispersal of information for security, load balancing, and fault tolerance. *J. ACM* **1989**, *36*, 335–348.
9. Li, M. On the Confidentiality of Information Dispersal Algorithms and Their Erasure Codes. *arXiv* **2013**, arXiv:1206.4123v2.
10. Jaeger, B. Security as a Service Working Group, Defined Categories of Security as a Service (Preview)—Continuous Monitoring as a Service. Cloud Security Alliance 2016. Available online: https://downloads.cloudsecurityalliance.org/assets/research/security-as-a-service/csa-categories-securities-prep.pdf (accessed on 10 August 2016).
11. Wall, M. Can We Trust Cloud Providers to Keep Our Data Safe? Available online: http://www.bbc.com/news/business-36151754 (accessed on 29 April 2016).
12. White Paper. Securing Sensitive Data within Amazon Web Services Ec2 and Ebs: Challenges and the Solutions to Protecting Data within the AWS Cloud. Copyright 2013 Vormetric. Available online: http://go.thalesesecurity.com/rs/480-LWA-970/images/wp-securing-data-within-AWS.pdf (accessed on 29 April 2016).

13. Rancourt, C. Celebrities Hacked: Are Your Personal Photos Safe in the Cloud? Available online: http://www.nextadvisor.com/blog/2014/09/02/celebrities-hacked-personal-photos-cloud-safe/ (accessed on 13 January 2017).
14. Ahmadi, M.; Moghaddam, F.F.; Jam, A.J.; Gholizadeh, S.; Eslami, M. A 3-level re-encryption model to ensure data protection in cloud computing environments. In Proceedings of the IEEE Conference on System, Process & Control , Kuala Lumpur, Malaysia, 12–14 December 2014.
15. Surv, N.; Wanve, B.; Kamble, R.; Patil, S.; Katti, J. Framework for client side aes encryption technique in cloud computing. In Proceedings of the IEEE International Advance Computing Conference, Banglore, India, 12–13 June 2015; pp. 525–528.
16. Singh, S.; Kumar, V. Secured user's authentication and private data storage-access scheme in cloud computing using elliptic curve cryptography. In Proceedings of the 2015 2nd International Conference on Computing for Sustainable Global Development, New Delhi, India, 11–13 March 2015.
17. Raj, G.; Kesireddi, R.C.; Gupta, S. Enhancement of security mechanism for confidential data using aes-128, 192 and 256 bit encryption in cloud. In Proceedings of the 2015 1st International Conference on Next Generation Computing Technologies, Dehradun, India, 4–5 September 2015.
18. Arockiam, L.; Monikandan, S. Efficient cloud storage confidentiality to ensure data security. In Proceedings of the International Conference on Computer Communication and Informatics, Coimbatore, India, 3–5 January 2014; pp. 1–5.
19. Suthar, K.; Patel, J. EncryScation: A novel framework for cloud iaas, daas security using encryption and obfuscation techniques. In Proceedings of the 2015 5th Nirma University International Conference on Engineering (NUiCONE), Ahmedabad, India, 26–28 November 2015.
20. Mar, K.K.; Law, C.Y.; Chin, V. Secure personal cloud storage. In Proceedings of the 10th International Conference for Internet Technology and Secured Transactions (ICITST-2015), London, UK, 14–16 December 2015.
21. Zhang, X.; Wang, H. A study of the use of idas in cloud storage. *Int. J. Future Comput. Commun.* **2013**, *2*, 67.
22. Rich, S.; Gellman, B. NSA Seeks to Build Quantum Computer That Could Crack Most Types of Encryption. Available online: https://www.washingtonpost.com/world/national-security/nsa-seeks-to-build-quantum-computer-that-could-crack-most-types-of-encryption/2014/01/02/8fff297e-7195-11e3-8def-a33011492df2_story.html?utm_term=.217ad6b56479 (accessed on 14 July 2016).
23. Balamurugan, S.; Sathyanarayana, S.; Manikandasaran, S.S. ESSAO: Enhanced security service algorithm using data obfuscation technique to protect data in public cloud storage. *Indian J. Sci. Technol.* **2016**, *9*, doi:10.17485/ijst/2016/v9i17/90229.
24. Smith, C. PointThe Year of BYOE (Bring Your Own Encryption). Available online: https://cipherpoint.com/2014/06/2014-the-year-of-byoe-bring-your-own-encryption/ (accessed on 15 June 2016).
25. Finite Field. Available online: https://en.wikipedia.org/wiki/Finite_field (accessed on 8 February 2017).
26. Minowa, T.; Takahashi, T. *Secure Distributed Storage for Bulk Data*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 566–575.
27. Rubenking, N.J. The Best Encryption Software of 2017. Available online: http://www.pcmag.com/article/347066/the-best-encryption-software-of-2016 (accessed on 5 June 2017).
28. Mishra, B.; Jena, D. Securing files in the cloud. In Proceedings of the 2016 IEEE International Conference on Cloud Computing in Emerging Markets, Bangalore, India, 19–21 October 2016.