Science Journal of Energy Engineering
2018; 6(1): 8-17
http://www.sciencepublishinggroup.com/j/sjee
doi: 10.11648/j.sjee.20180601.12
ISSN: 2376-810X (Print); ISSN: 2376-8126 (Online)

Security of Cloud Virtualized Resource on a SaaS Encryption Solution

Chinedu Paschal Uchenna, Nwankwo Wilson

Department of Computer Science, Wellspring University, Benin City, Nigeria

Email address:
pchinedu@wellspringuniversity.edu.ng (C. P. Uchenna), wilson.nwankwo@wellspringuniversity.edu.ng (N. Wilson)

To cite this article:
Chinedu Paschal Uchenna, Nwankwo Wilson. Security of Cloud Virtualized Resource on a SaaS Encryption Solution. *Science Journal of Energy Engineering.* Vol. 6, No. 1, 2018, pp. 8-17. doi: 10.11648/j.sjee.20180601.12

Received: February 22, 2018; **Accepted**: March 9, 2018; **Published**: April 3, 2018

Abstract: Protection of user data against data breaches in cloud applications and the potential security failures of the service providers coupled with heightened cloud user apprehension, have in no small degree defied measures taken to demystify cloud services as to unveil its enormous capacity and awesome benefits such as accessibility, availability, collaboration, to name a few. The security of the cloud infrastructure entails protecting cloud data from unauthorized access, preventing malicious programs from corrupting the virtual resource and ensuring the secure cloud data remains unintelligible to any unauthorized access or intrusion by malicious users. This paper is aimed at building a cryptographically secure cloud application environment. Its major objective is to design and implement an encryption system for protecting valuable data (such as passwords, messages, files) in the cloud environment. The design and implementation extended some basic security and privacy requirements including data confidentiality, integrity, and availability by considering fairness as a viable factor. This paper employed the Structured Systems Analysis and Design Methodology (SSADM) in the software development life. It evolves a novel cryptographically-secure cloud algorithm based on a proposed "Deciv Algorithm" tagged "D65- Enc" algorithm that would effectively hide meaningful user data from all external parties to a virtual network as well as the service provider by putting control in the hands of users. The algorithm is carefully crafted to frustrate any cryptanalyst, hacker or cybercriminal who would try to decipher the algorithm. This implementation is expected to assist cloud users in maintaining control over their data whether at rest or in transit within the cloud networks rather than outsource control to external vendors as usual. Moreover, this algorithm also improves the existing state of data privacy, and security in the cloud.

Keywords: Cloud Computing, Cloud Security, Encryption, Algorithms

1. Introduction

For many decades, encryption has been used as a security measure to render data unintelligible to unauthorized parties. However, its popularity and need has increased as social and community computing platforms (such as the cloud) are continuously evolving in modern business and social relationships. In any case, data have remained the most important resource to the user, and in a public cloud where communal computing and multitenancy are practiced, encryption is inevitable to ensure confidentiality and integrity of the transmitted data and as well as the data bank. Encryption is implemented with the objective of curbing impersonation, wiretapping, piracy, spoofing and data diddling, etc. which are common forms of privacy evasions and abuses in a multi-user environment. Before now, there has been Blowfish, Rijndeal, AES, Python, Crypt and so many other encryption algorithms presently in existence to combat some of these known threats. However, the said objective has not been achieved as Hackers are out with their botnets, rainbow tables, and other means aimed at decrypting all the known encryption algorithms. Users with confidential data are gripped with fear of insecurity, even the service providers are not sure of the data security despite the encryption used. And where data security measures are implemented by cloud providers, users are not sufficiently assured sole ownership of control of their useful, confidential or classified sensitive data. This challenge which raised

questions within the constituencies of consumers of cloud services is the central theme of this paper.

1.1. Statement of the Problem

The security solutions such as: the use of secure authentication, authorization, and identity management, and the securely encrypted and integrity-protected user data significantly impact the security solutions among the constituencies of the various cloud consumers but fail to contain the concerns on the privacy and control of user data and as such inferring notable deterrent to cloud adoption. Previous researches had propounded models which affirmed that the secure authentication, authorization and identity management, and the securely encrypted and integrity-protected user data had notable negative effect on cloud data security. The concern here borders on dependability and effectiveness of the existing security amidst the outsourcing of the control of users' sensitive data via the cloud to a third party cloud service provider. This concern, amidst the ever-increasing use or dependency on virtualized resource and multi-tenancy motivated the need for a dependable and secured software as a service (SaaS) security solution of this research.

1.2. Specific Objectives

The specific objectives of this paper are:

1. To design and implement an Encryption System for securing valuable data (such as authentication credentials, messages, and files) in the virtualized cloud environment.
2. To emphasize fairness as part of the security and privacy requirements of data confidentiality, integrity, availability;

1.3. Cloud Computing Security

The increasing adoption of the virtualized environment have resulted in the recent rise in cloud computing. Consequently, the security of the virtual environment is now the primary concern of cloud security [1]. Nolle [2] has argued that the Virtual Environment does have some pitfalls, but none are so severe enough as to eliminate it entirely. This security is the foundation upon which the service (including Software as a Service (SaaS)) lies. In the investigation, there are several areas of concern regarding the virtual environment and the ability to provide sufficient security [1].

1.4. SaaS Cloud Model

Understanding the issues in cloud computing is better grasped through adequate discussion of the three computing models that come under its canopy. These models also described as delivery models [3] are:

1. Software as a Service – SaaS
2. Platform as a Service – PaaS
3. Infrastructure as a Service – IaaS [4]

In SaaS the infrastructure has the software applications and enterprise systems installed by the cloud service provider; the customer uses web browser or other internet technologies for accessing the hosted cloud applications [3]. Unlike traditional applications that users install on their computers or servers, Warr [5] emphasized SaaS software is owned by the vendor and runs on computers in the vendor's data center (or a colocation facility). The customer's control over any aspect of the hardware, software, and the security for both does not apply – sometimes the only possible control extended to the customer is in the maintenance of the security for the instance of the software being used or accessed [3]. Generally speaking, all customers of a SaaS vendor use the same software: these are one-size-fits-all solutions. Well known examples are Salesforce.com, Google's Gmail and Apps, instant messaging from AOL, Yahoo and Google, and Voice-over Internet Protocol (VoIP) from Vonage and Skype [5].

1.5. Cloud Data Security Concerns

Samson [6] in an RSA Conference of Cloud Security Alliance (CSA) showcases cloud vendors copiously hawking products and services that arm IT with controls to foster order to the threatening cloud chaos. The exercise requires for organization to identify and rank the greatest cloud related threats and where they occur. The report reveals the unanimous consensus among industry experts, focusing on shared, on-demand nature of cloud computing related threats which gave highest priority to data breaches such as data loss and data leakage.

Earlier work by IBM Research [7] highlights five key concerns about cloud computing. These gave foremost priorities to the issues of "less control" of user to their own data, and Data Security in a shared network and compute infrastructure to unveil threats on and discomfort by many companies and governments with the concept of locating data on systems not user controlled and the increasingly potential for unauthorized exposure in multi-tenants environment respectively. The research further addressed these concerns by proffering the following solutions:

Less Control: Provider become fully security transparent and offer sophisticated control

Data Security:

a) Implement secure authentication, authorization, and identity management
b) Isolate multiple tenants from each other
c) Encrypt critical data and ensure they are integrity-protected by client.

Furthermore, Frye [8] describe a new open source project called *CRYPTON*, which development was said to be in-progress, which is hoped to put a reusable cryptographic solution in the hands of cloud application developers by providing easy, built-in encryption of user data.

Violino [9] has listed thirteen (13) cloud computing security concerns which are as follows:

i. Data breaches- these may involve data not intended for public access such as personal health data, financial data, trade secrets, intellectual property, etc.
ii. Insufficient identity, credential, and access

management- whereby hackers may masquerade as legitimate users, developers, etc. to read, modify, and delete data; issue control plane and management functions; snoop on data in transit or release malicious software that appears to originate from a legitimate source, CSA says.

iii. Insecure interfaces and application programming interfaces (APIs)- Cloud providers often expose APIs that enable customers to integrate interfaces that can communicate with cloud services. Since provisioning, management, and monitoring are conducted using the said APIs, the security of cloud services would inherently depend on the security of APIs.

iv. System vulnerabilities- These are exploitable bugs in programs that attackers may exploit to infiltrate a system so as to steal data, take control of the system, disrupt service operations.

v. Account hijacking- Attackers may gain access to a cloud user's credentials by eavesdropping on activities, transactions, data, etc.

vi. Malicious insiders- A malicious insider such as a system administrator can access potentially sensitive information, and can have increasing levels of access to more critical systems and eventually to data. Systems that depend solely on cloud service providers for security are at greater risk.

vii. Advanced persistent threats (APTs)- APTs are a parasitical form of cyber attack that infiltrates systems to establish a foothold in the IT infrastructure of target companies, from which they steal data. APTs pursue their goals stealthily over extended periods of time, often adapting to the security measures intended to defend against them. Once in place, APTs can move laterally through data center networks and blend in with normal network traffic to achieve their objectives

viii. Data loss-Data stored in the cloud can be lost permanently through accidental deletion by the cloud service provider, physical catastrophes (e.g. fire, earthquake, flood, etc.)

ix. Insufficient due diligence -Business strategies must as a matter of due diligence take into consideration the cloud technologies and service providers. Available technologies and providers should be sufficiently evaluated so as to avoid a number of risks associated with low credibility and service delivery indexes.

x. Abuse and nefarious use of cloud services- It is submitted that insecure or poorly secured cloud deployments, free cloud trials, and fraudulent account sign-ups via payment instrument fraud expose cloud computing models to malicious attacks. Attackers would often leverage on the said loopholes to target users, organizations, or other cloud providers. For instance, an attacker on gaining access to a cloud-based resource can launch distributed denial-of-service attacks, email spam, and phishing campaigns.

xi. Denial of service (DoS) - DoS_attacks often prevent legitimate users of a resource from gaining access to the resource. Attackers can cause a shutdown of a cloud resource by bombarding a targeted cloud service with inordinate amounts of transactions which take up much resource such as processor power, memory, disk space, or network bandwidth, etc.

xii. Shared technology vulnerabilities- scalability is important in most cloud service delivery operations. In other words the service providers often share infrastructure, platforms or applications at the expense of security in most cases.

xiii. Spectre and Meltdown- these two threats are mostly a lapse associated with modern microprocessors used in mobile devices, PCs, servers, Cloud, etc. whereby content, including encrypted data could be read from memory using malicious Javascript code. Meltdown breaks the isolation between user applications and the operating system thus allowing a program to access the memory including confidential credentials of other programs and the operating system. Spectre, on the other hand, breaks the isolation between different applications permitting an attacker to trick error-free programs, which follow best practices, into leaking their secrets.

1.6. Data Encryption

Encryption is an information security measure that renders data unintelligible to unauthorized readers. It is a coded transformation of data into a form unreadable to intruders and interlopers who lack the appropriate key to decrypt the encoded data [10]. Encryption involves using a cryptographic algorithm and a cryptographic key in order to transform a plaintext into a ciphertext or not obvious text [11]. Figure 1 gives us diagrammatical illustration of a basic encryption system.

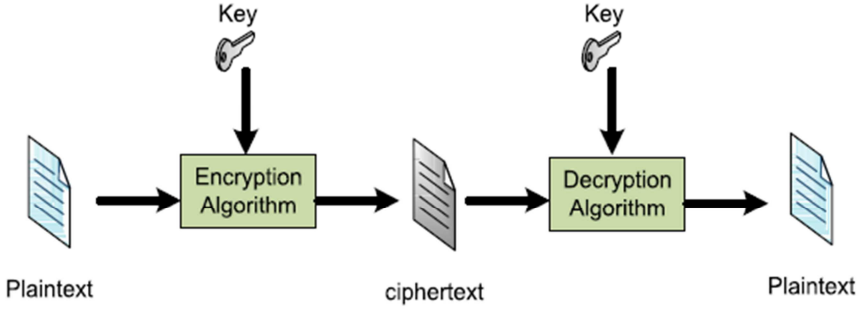

Figure 1. *Basic Encryption (Al Beshri, 2013).*

Encryption is gaining popularity as social and community computing (such as the cloud) is gaining momentum. Encryption Technique is important not just for the data but also for database controls and communication channels such as the Secure Socket Layer (SSL). In a public cloud where communal computing and multi-tenancy is practiced, encryption must be inevitable to ensure confidentiality and integrity of information and data store. As a mitigation technique that could sufficiently address the risk of information disclosure threat type, it is believed that the essence of encryption is to curb impersonation, wiretapping, piracy, spoofing and data diddling which are common abuse in a multitenant environment.

The two most important techniques for encryption are the Symmetric Encryption and Asymmetric encryption. Both are vital and applicable as a hybrid encryption for secure implementation of a public cloud.

1.7. Symmetric Cryptography

This is a private-key encryption technique that is based on shared secrete between the two communicating parties [12]. This party generates a key that allows them to encrypt and decrypt messages and the key is kept secret. No one can read the message except with the key (Electronic Frontier Foundation, 2004). The key is just an algorithmic seed (called $Salt in PHP) that will turn data into rubbish. That same 'seed' is required to turn the rubbish into the original data [13].

Single user, such as an organization in a cloud can choose a particular key, use it to encrypt his information base and keep the key very secret. Such information-base will continue to be confidential until the organization let the key out to someone else or intruders hacked-in and retrieve the key.

1.8. Asymmetric Cryptography

This is a public key encryption that is based on matched key-pair. Paar *et a*l [12] argued that the participant to such secure communication generates two keys. One portion is made private and kept secret, while the other is published to the public. The public can use the key to decrypt any information encrypted with the private key-pair. Although this is called public, yet it's only those that have the key-pair (public) that can decode and read the ciphered information. In a public cloud, service provider can have a cloud-wide public key which the cloud- tenants can use to decrypt information within the cloud environment while outsiders and intruders cannot read their information since they do not have the key. This will also make it easy to detect foreign and fabricated information because the cloud-public key cannot decrypt the foreign information injected into the platform by interlopers and cyber criminals.

1.9. Hybrid Encryption Scheme

A Hybrid cryptosystem can be used to crypt and decrypt both private and public keys. Such systems can be used for key distribution and to encrypt bulk data with high speed [14]). This scheme is used today to secure web-based transactions as well as secure email services and other communication systems such as Netscape communicator, secure socket layers and digital signatures. A typical example of such system is the R.S.A machine.

2. Existing and Proposed Solutions

The security of cloud data had been identified as the main obstacle to implementing effective cloud computing platforms. In addition to security, other issues revolve around ompliance, and legal issues including liability, intellectual property, etc. Among other areas of risk (such as dependency on the public Internet, multi-tenancy and integration with internal security) on the move to cloud, external data storage receives the most attention [15]. Most Organizations are indeed worried about security and privacy concerning the use of cloud computing services as there is negligible assurance coming from the market. Matching internal user security requirements with the security measures and controls employed by cloud computing vendors or service providers such as authentication and identity management, and data encryption proves to be impracticable due to discrepancies, lack of transparency and insufficient expertise inhibiting the trust from most cloud users and VE participants.

With the -0.296 contribution of 'authentication and identity management', and -0.358 impact of 'encryption and integrity-protected user data' on the existing security solution in a hypothetically Cloud Security model (i.e. Cloud Sec = 3.371 – 0.296AUTH – 0.359ENC) [14], there are still major and notable concerns to data security in the cloud due to failure of service providers and malicious attacks from hackers, amidst the widespread eagerness on cloud computing deployment. The reoccurring event of data breaches of significant cloud services experience by many still caused some decline in the enthusiasm especially among potential customers with extremely sensitive data from the plans to manage and move confidential resources into the cloud. Besides, the enormous security benefits accruable from the cloud vendors' services, the respondents to the interview in the research, argued on the matter of trust on a third party provider, and the risks of outsourcing control of their data. Therefore, the cloud is inherently neither secure nor dependable from the perspective of some of the cloud customers [14]. Where no strong, dependable and trusted security and privacy measures are provided to assure on data confidentiality, integrity, availability and fairness, expecting cloud customers to turn over their data as well as the control of their data to a third party cloud and virtualized infrastructure solely based on quest of leveraging on the economic and security service benefits, or reducing corporate IT expenditure would be a defeated purpose.

The Cloud Standards Customer Council [16] has prescribed a ten-step solution to ensuring cloud data security:

i. Ensure effective governance, risk and compliance processes exist
ii. Audit operational and business processes
iii. Manage people, roles and identities
iv. Ensure proper protection of data and information
v. Enforce privacy policies
vi. Assess the security provisions for cloud applications
vii. Ensure cloud networks and connections are secure
viii. Evaluate security controls on physical infrastructure and facilities
ix. Manage security terms in the cloud service agreement
x. Understand the security requirements of the exit process

It is in view of the need for proper protection of data that this paper evolved a novel SaaS data security solution. The methodology used was the Structured Systems Analysis and Design Methodology (SSADM).

2.1. The Proposed SaaS Encryption Solution

Table 1 shows a comparison between the existing private encryption implementation and the proposed solution as may be applied to a SaaS cloud solution. Similarly, Table 2 shows a comparison between the existing public encryption and the proposed solution.

Table 1. *Comparing the existing private encryption and the proposed solution for SaaS cloud solution.*

Existing system	Proposed solution
Have recognizable head in the cryptext which gives the hackers knowledge of the Algorithm that formed the cryptext leading to the eventual breaking of the encryption given time and resources. The existing systems mostly use static Initialization Vector (IV) which is subject to guessing and adversely affects the robustness/hardness of the encryption.	Generates a cryptext (ciphertext) with a pseudo-head (false head) that is deceptive to the hacker. The cryptext appears to use Blowfish Algorithm while in the real sense it is Rijndael. This therefore mis-leads the hackers.

Table 2. *Comparing the existing public encryption and the proposed solution for SaaS cloud solution.*

Existing system	Proposed solution
Uses a decryption key made public (Asymmetric) to all intended users and is given by certification authorities and had validity period before expiration. Its disadvantage is that since it's a public knowledge, it's no longer secret, and a bias certification authority personnel can leak the secret (key) to unauthorized users.	Encrypted so that it's meant for organized users or community e.g VO. The encryption doesn't expire with time as it's generated within the code, not by certification authorities. Since the Initialization Vector (IV) is randomly generated and internally encrypted, it cannot be guessed or exposed.

2.2. The Proposed Customized Algorithm for Cloud Services

Every encryption algorithm has a uniform pattern in the cyphertext with which it is recognized. Crypto-analysts can build a rainbow table to decrypt the cyphertext even if it takes a long time. Consequently, no encryption is foolproof except the "Deciv Algorithm". The concept of "Deciv Algorithm" is to use salty encryption algorithms but removed the uniform pattern with which the algorithm can be identified and replace it with something else. The objective is to make the cyphertext unrecognizable to hackers so that no Rainbow table can be built for it and thus it remains undecipherable. Two Algorithms are proposed. One is a public key encryption which runs on the platform and encrypts very confidential data including virtual databases from intruders. However every tenant of the cloud will have the decrypting key. The other is a private key encryption which only the user knows and keeps secret. Each of these encryptions does not follow universal open-source algorithms which the hackers can identify, which has a greater than zero probability to be in the rainbow table, which has a potential of being broken someday; rather the proposed encryption algorithms are products of series of encryption sequence and transformations using PHP String functions.

There has been Blowfish, Rijndael, AES, MD5, Crypt and so many other encryption algorithms presently in existence to combat some of these known threats, such as impersonation, wiretapping, piracy, spoofing and data diddling, which are common abuse in the cloud or a multi-user environment.

In developing the cryptographically secure cloud data environment tagged "Pablo Cloud-Based Encryption System", the novel algorithms within the encryption system has been tagged "D65-Enc". The algorithm uses "Deciv value" which defines the D in the adopted name. The two main modules of the algorithm contained in the system are:

The Public (Platform) Encryption (with included Salt- Code)
a. Create a function for the encryption requesting user data only
b. Randomly Generate an Initialization Vector (IV)
c. Invoke the Rijndael_128 Mcrypt as core
d. Make the "deciv" adjustments
e. Output the "deciv" value as cryptext (ciphertext)

The Private Encryption with User Defined Salt- Code
a. Create a function requesting user data & user salt-key
b. Invoke the AES crypt method as core
c. Reverse the content of user data before encryption
d. Make the "deciv" adjustments on the encrypted data
e. Output the "deciv" value

2.2.1. The Proposed Private Key Encryption

This symmetric Encryption uses the AES-256 encryption (known as Rijndael Algorithm) as the core, with modifications made to shield its identity from Hackers and also to make it more robust. The salt is provided by the users so that they can be rest assured of the security of their information. The Information to be secured is first serialized, to generate a storable representation of the information. Next an Initialization Vector (IV) is created to give alternative seed to the encryption routine. The IV here is randomly generated by the Computer to initialize the CBC (Cipher Block Chaining) Mode.

The Output (Cyphertext) is appended with a $2y$31$ String. The appendage will not only nullify Oracle padding attacks by confusing the Oracle, but will also

deceive the hackers into thinking that the encryption is a blowfish Algorithm. It will take the Hacker time, energy and resources to discover (if ever he can) that the encryption is an AES-256, even at that; it will take over a million years to crack the AES-256 encryption. This algorithm is shown below in figure 2:

```
<?php Define („ENCRYPTION_KEY?, $user_salt);
Function mc_encrypt ($encrypt, $key){

$encrypt = serialize ($encrypt);
$iv = mcrypt_create_iv(mcrypt_get_iv_size(
MCRYPT_RIJNDAEL_256, MCRYPT_MODE_CBC), MCRYPT_DEV_RANDOM);
$datacrypt = mcrypt_encrypt(MCRYPT_RIJNDAEL_256, $key, $encrypt,
MCRYPT_MODE_CBC), $IV);
$code_it = base64_encode ($datacrypt).?7?.base64_encode ($iv);
$deciv = „$2y$32$?. $code_it;
Return $deciv;
    }
```

Figure 2. *Algorithm for a private encryption solution for cloud service.*

2.2.2. The Proposed Public Key Encryption Solution

The engine of the public key encryption is the Blowfish using salt type $2y$ and hacker slow number of 22. BCrypt is a one-way Hash Algorithm, meaning that it cannot be decrypted. The Encryption uses a 62-digit randomly generated salt to ensure the cyphertext is not decipherable. However, after the encryption, the Cyphertext is subjected to PHP Ltrim() function to cut off the harbinger $2y$22$ which shows the hackers the type of Algorithm used. So by cutting off this $2y$22$, the Hackers will not be sure of the hash algorithm used. Who knows, they might be fooled forever. It is impractical to decrypt a Cyphertext without knowing the algorithm that produced it. The proposed algorithm is shown in figure 3.

```
<?php
//first we declare the Algorithm as a function for reusability sake.
// the function requires two inputs: the Value to encrypt and the rounds number of hardness.
Function code_public($input_Val, $rounds){
$salt = " ";
//the declared salt is filled with the output of 22 random Numbers coined from Alphanumerics
$salt_chars = array_merge(range(„A","Z"),range(„a","z"),range(0,9));
For ($i=0; $i<22;$i++){
$salt. = $salt_chars[array_rand($salt_chars)];
}
//the First „Deceive" step is taking by reversing the PlainText before encrypting it.
$rev = strrev($input_val);
$hash = crypt($rev, sprint(„$2y$22$", $rounds).$salt);
//the „Decive" Output is done by trimming off the Prefix indicator of the Algorithm.
$deciv = ltrim($hash, $2y$22$);
Return $deciv;
}
```

Figure 3. *Algorithm for a public encryption solution for cloud service.*

It is not unreasonable to believe that no encryption, no matter how salty it tastes, will remain undecipherable forever, so long as the algorithm is know; it is only a matter of time before the malicious cryptanalysts will see through it. That‟s why we have chosen the „Deciv Algorithm‟ The *Deciv algorithm* cannot be fathomed by automatic machines, only human reasoning might decode the deception, yet it must require extra- reasoning and ultra-high amplitude of perception to achieve that, in fact the tendency approaches zero. Meanwhile nowadays, only automated systems are used such as Brute Force, Rainbow table, The Padding Oracle and others; therefore this algorithm is one of the most brute resistant encryption ever to be used by cryptanalysts and hackers.

3. Overall Dataflow Diagram of Proposed Solution

Figure 4 shows the model of a cryptographically-secure cloud data environment solution. The said model is represented using a Data Flow Diagram (DFD) which is suitable for analyzing and constructing information processes. Also known as a Process Model, the DFD is an illustration that explains the course or movement of information in a process-oriented environment. Figure 4 shows how data originating form a primary source usually at the cloud user's computer travels across a public network such as the Internet either

encrypted or unencrypted. Where the data to be stored in the cloud is controlled at the originating end, solid blue arrows represent encrypted data whereas a dotted red arrow represents unencrypted data flows.

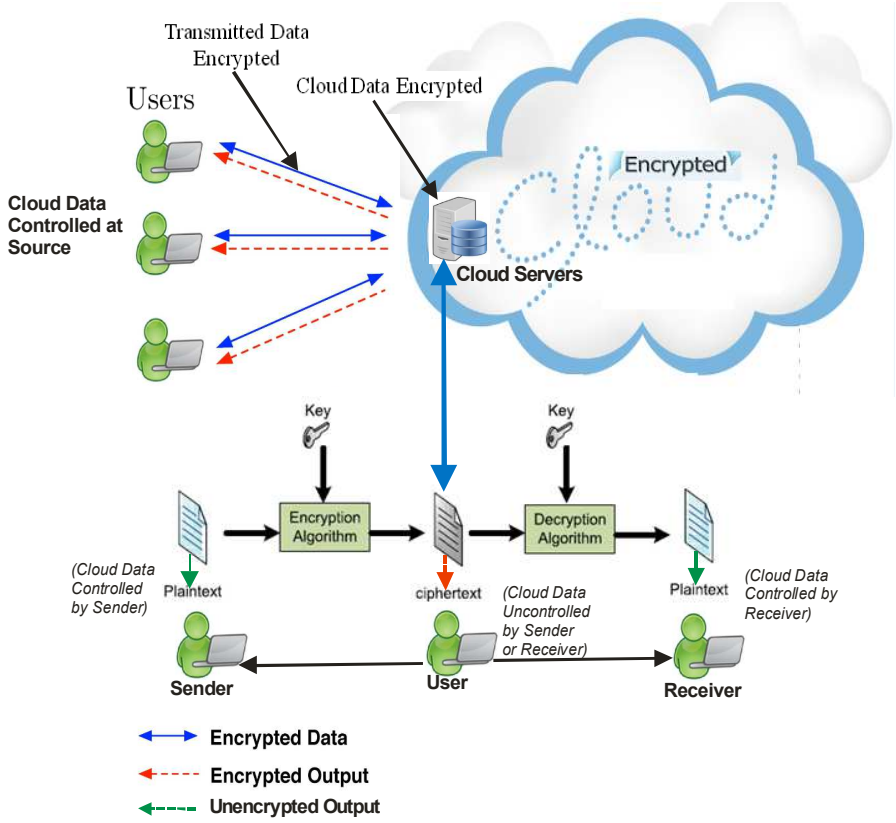

Figure 4. *Dataflow Diagram of the proposed encryption (source; Chinedu, 2014).*

4. Implementation

The implementation phase involves the interaction of the coding, testing and the integration stages of the various software components to deliver the security system. The cloud encryption system is a cloud-based SaaS application with full data security and privacy measure in place for the cloud application service environment. The result of the implementation is a fully functional cloud-based encryption system platform accessible via a web browser in an Internet ready computer system.

4.1. Minimum System Requirements

The implementation is cloud-based, i.e. is meant to run on virtualized cloud infrastructures which would dynamically scale up or down on demand to ensure optimum systems performance. The minimum specifications for the cloud server are:

a. Available speed of 2.5GHZ
b. 4GB RAM
c. 30GB Hard disk
d. High Speed Network Interface Card
e. Uptime of 99.9% or more
f. 1Gbps Bandwidth

The minimum specifications for the user's system are: A Pentium 950MHz system with at least 256MB RAM running Microsoft windows XP or latter, 6.7mbps Modem or Broadband service, and a minimum storage space of 10GB on Hard Disk.

4.2. Choice and Justification of Programming Language

The language platform of choice is PHP. PHP is a w3 standard, an open source and most importantly, a cross platform. That means it can run in different platforms such as Windows, Linus, UNIX, etc. Moreover, PHP has many built-in functions and global variables that are predefine for encryption and data security through encryption, example is MD5. PHP allows for user defined functions which makes the platform very flexible for additional functions and plugins to the platform leading to innovation supports. PHP serves as the middleware between Users and the database thus is a good agent for bridging and checking user status before granting access to the database and also reading and authenticating user request for appropriate output. Rijndael Encryption Standard is chosen. Rijndael is about the world's number one encryption standard in terms of robustness, simplicity, reliability and Salty Security. Reports through Practical test have it that it will take cryptanalysts (hackers) 2 million years to break Rijndael encryption; another way of saying it cannot be broken. Rijndael is Salty. The salty nature makes it almost 100% secure, Rainbow table and brute attack proof.

4.2.1. The Private Key Encryption Process

This is a symmetric Encryption that uses the AES-128 encryption as the core, with modifications made to shield its identity from Hackers and also to make it more robust. The salt is provided by the users so that the security of their information. The Information to be secured is first serialized, to generate a storable representation of the information. Next an Initialization Vector (IV) is created to give alternative seed to the encryption routine. The IV is randomly generated to help harden the encryption and negate the effect of guessing or any other brute-force attack method. The Output (Cyphertext) is appended with a $2y$22$ String. The appendage will not only nullify Oracle padding attacks by confusing the Oracle, but will also deceive the hackers into thinking that the encryption is a blowfish Algorithm. It will take the hacker time, energy and resources to discover (if ever possible) that the encryption is an AES-128, even at that; it will take over a million years to crack the AES-128 encryption.

4.2.2. The Public Key Encryption Process

The engine of the public key encryption is also a Rijndael encryption engine with a randomly generated Initialization Vector (IV). Mcrypt- 128 is fed with the IV which serve as the encryption seed. The input to the encryption is the user data obtained from the platform whenever the "Send" button is clicked without supplying a salt code. Every data sent through and within the platform is thus encrypted such that even if hackers invade the platform and pilfer some data, they cannot read it. The platform encryption occurs without the authorized user's consent as the users read their data as if it was not encrypted but to external users (unauthorized users) the data from the platform appears and is encrypted. The hardness of the encryption is fortified by the 128-bit encryption IV that is randomly generated. It is not unreasonable to believe that no encryption, no matter how salty it tastes, will remain undecipherable forever, so long as the algorithm is know; it is only a matter of time before the malicious cryptanalysts will see through it. That's why the 'Deciv Algorithm' tagged "D65-Enc" is preferred. The Deciv algorithm cannot be fathomed by automatic machines, only human reasoning might decode the deception, yet it must require extra- reasoning and ultra-high amplitude of perception to achieve that, in fact the tendency approaches zero. Meanwhile nowadays, only automated systems are used such as Brute Force, Rainbow table, The Padding Oracle and others; therefore this algorithm is one of the most brute resistant encryption ever to be used by cryptanalysts and hackers.

4.3. Setup Procedure

The entire encryption system was implemented on a local machine. There is no special setup required to run this program on a fully Internet ready platform. To Setup the program, the following procedure will suffice:

1. Buy a domain name from an Internet Service Provider.
2. Obtain a Control panel account on the chosen domain that supports PHP and MySQL.
3. Upload the program files to a public folder and activate the index.php file.
4. View the site using a web browser.

4.4. System Testing and Evaluation

To encrypt information (such as messages and passwords) in the cryptographically- secure cloud data environment of the "Pablo Cloud-Based Encryption System", the following test and evaluation would suffice:

1. *Procedure:* Testing messages selected:

A. A message in form document "sent without salt code included)" at source. When message reaches the receiver, no salt code required to open the readable document. Here *public key* was enforced where appropriate authentication privilege was granted at login by the registered user (receiver). *Platform encryption* occurred at the backend (database) in case of unauthorized access to the database.

B. A message in form document "Sent and encrypted (with salt code included)" at source. When message reaches the receiver the salt code (*private key*) would be needed to open the document to make it readable. *Private encryption* occurred on sent message and at the backend (database) in case of both authorized access without appropriate salt code to view the message, and unauthorized access to the application or database.

2. *Test:* A document (message) was composed and sent from source to the receiver in line with the above procedures.
3. *Result:* On receipt by receiver, the message was made unintelligible and the receiver:

A. With successful login could access and view the readable message without salt code required. Any unauthorized access (with failed authentication at login) to database views the message as "platform" encrypted.

B. With successful login required (requested) a private key from the source which was sent to the receiver. On application by receiver, the document opened and was readable.

4. *Evaluation:* This fulfilled the expectation of the paper.

Tables 3-4 shows sample tests and expected outcomes from the implemented algorithm.

Table 3. *Public Encryption Test.*

Test	Outcome	Comment
User saves data	stored encrypted in database	Ok
User request data	retrieves unencrypted	Successful Authentication
Outsider request data	retrieves encrypted	None; Register User

Table 4. *Private Encryption Test.*

Test	Outcome	Comment
User saves data with slat key	stored encrypted in database	Ok
User request data with correct key	retrieves unencrypted	Auth.+ Salt
User (Hacker) requests data	retrieves encrypted	Salt code required
User (Hacker) enter the wrong key		Wrong Key

4.5. Application of the Proposed Solution

The application of this study cuts across the enterprise, industry and public policy levels. The SaaS based cryptographically secure cloud apps environment tagged "Pablo Cloud-Based Encryption System" has industry wide implications, and although we have not attempted to deploy the app for any of the security concerned organizations in the course of the field survey or an industry, we believe that it would adapt with any state of the art cloud apps in the industry in respect to its potential capability to operate collaboratively or its amenability to current information technology haven being implemented and tested on an Internet hosting platform. Cryptographic techniques have been used in this research to protect data before it is moved to the cloud. The Encryption tools could be deployed by cloud users, participators of a virtual organization and other virtual world to secure their data at rest (stored) and on transit (when being transmitted) or their identity management in any advanced ICT network. This security service is a solution that contains the issues of data availability- granting access to only authorized user, data confidentiality- ensuring only authorized user can access the data and data integrity- such that any modification to the data could be detected.

Most security solutions are built and managed through the expertise and administration of security personnel who usually work at the ends of the cloud service provider. Thus, users remain sensitive to the privacy level of their data as control remains outsourced. Therefore, this solution would gain outstanding applicability because most cloud users are becoming "privacy aware. Also, enterprises are refusing to adopt solutions that keep data control in the hand of providers, where the developers and these providers can access their critical internal data.

5. Conclusion

This paper upon investigating security weakness in the adoption of cloud model for corporate value creation following various security and privacy issues provides an improvement on the existing security solution. The proposed solution could be implemented by service providers, organizations or consumers or as a SaaS application when outsourcing data, applications, and infrastructure to a virtualized cloud environment. The security of the cloud infrastructure entails protecting the cloud data from unauthorized access, preventing malicious program from corrupting the virtual resource and ensuring the secure cloud data remains unintelligible to any unauthorized access or intrusion by malicious users. This paper propounded a novel cryptographically- secure cloud algorithm based on a proposed "Deciv Algorithm" tagged "D65- Enc" algorithm which would effectively hide meaningful user data from all external parties to a virtual network including the cloud service provider. The new system would improve the existing state of data privacy and security in the cloud.

References

[1] Sample, C. (2012) "IaaS security puts spotlight on hypervisor security, tenant Management" [Online]. Available from http://searchcloudsecurity.techtarget.com/tip/IaaS-security-puts-spotlight-on-hypervisor-security-tenant-management [Accessed: August 16, 2012]

[2] Nolle, T. (2012) "Pros and cons of a non-VM-based IaaS model" [Online]. Available from http://searchcloudcomputing.techtarget.com/tip/Pros-and-cons-of-a-non-VM-based-IaaS-model [Accessed: 23 February, 2018]

[3] Reilly, D.; Wren, C. & Berry, T. (2011) "Cloud Computing: Pros and Cons for Computer Forensic Investigations", International Journal Multimedia and Image Processing Vol. 1, Issue 1

[4] Siddiqui, M. (2011). Cloud Computing Security: [Online] Final paper submitted spring 2011. Available from: http://blogs.techconception.com/manny/content/binary/Manny%20Siddiqui%20-%20Cloud%20Computing%20Security.pdf [Accessed: 20 May 2011]

[5] Warr (2009). Cloud computing. Available from http://www.qsarworld.com/files/Cloud-computing.pdf [Accessed: 14 August 2012]

[6] Samson, T. (2013) "9 top threats to cloud computing security. Conference processing by Cloud Security Alliance"[Online]. Available from http://www.infoworld.com/t/cloud-security/9-top-threats-cloud-computing-security-213428?page=0,0 [Accessed: 05/06/2014]

[7] IBM Research (2011) "Protocols for Secure Cloud Computing: Christian Cachin, Zurich" [Online]. Available from http://www.zurich.ibm.com/~cca/talks/metis2011.pdf [Accessed: 21 May 2013]

[8] Frye, S. (2013) "Crypton for developers: Toward cryptographically- secure cloud apps"[Online]. Available at: http://www.techrepublic.com/blog/linux-and-open-source/crypton-for-developers-toward-cryptographically-secure-cloud-apps/ [Accessed: 27/05/2014]

[9] Violino, B(2018) "The dirty dozen: 12 top cloud security threats for 2018"[online]. Available at: https://www.csoonline.com/article/3043030/security/12-top-cloud-security-threats-for-2018.html. [Accessed 7 March 2018]

[10] Hellman, M. E. (1980) "A cryptanalytic time-memory trade-off. Information Theory", IEEE Transactions, Vol. 26, Issue: 4

[11] AL Beshri, A. M. (2013) Outsourcing data storage without outsourcing trust in cloud Computing, A Thesis submitted in partial fulfilment of the Requirements of Queensland University of Technology for the Degree of Doctor of Philosophy. Available from http://eprints.qut.edu.au/61738/ [Accessed: June 05, 2017]

[12] Paar, C.; Pelzl, J. & Preneel, B. (2010) Understanding Cryptography: A Textbook for Students and Practitioners, Springer

[13] Graham, R. D. (2011). "Password cracking, mining, and GPUs"[Online]. Available from http://www.erratasec.com [Accessed: 17 August 2011]

[14] Ristic (2010) "Internet SSL Survey 2010 Black Hat USA" [Online]. Available from https://media.blackhat.com/bh-us-10/presentations/Ristic/BlackHat-USA-2010-Ristic-Qualys-SSL-Survey-HTTP-Rating-Guide-slides.pdf [Accessed: August 02, 2014]

[15] Chinedu, P. U. (2015) Modelling a Secured Cloud-based Framework for ICT intensive Virtual Organization. A Thesis submitted in partial fulfilment of the Requirement of Federal University of Technology, Owerri for the Degree of Doctor of Philosophy. (Unpublished)

[16] Cloud Standards Customer Council(2017) "Security for Cloud Computing: Ten Steps to Ensure Success"[online]. Available at: http://www.cloud-council.org/deliverables/CSCC-Security-for-Cloud-Computing-10-Steps-to-Ensure-Success.pdf. [Accessed 7 March 2018]