International Journal of Electrical Engineering.
ISSN 0974-2158 Volume 12, Number 1 (2019), pp. 1-20

http://www.irphouse.com

Balancing Security and Application Functionality in Cloud-based Applications: A Survey

Abiodun Esther Omolara*
Aman Jantan, Oludare Isaac Abiodun
School of Computer Sciences,
Universiti Sains Malaysia, 11800, Penang, Malaysia

Emmanuel Etuh,
Department of Mathematics,
Arthur Jarvis University, Nigeria

ABSTRACT

Cloud computing has risen as a prevailing computing platform for years to come, impacting the functionality, design and deployment of applications. As cloud computing advance in prominence, it becomes a major hit for attackers and it security issue a huge concern for practitioners. These security issues have given rise to several active researches in securing cloud-based applications which produced modern encryption schemes. These modern encryption schemes are considered secure enough to provide security to the cloud applications however at the expense of some functionalities of the application. Current research in the cloud-based domain is predominantly engaged with accomplishing optimal security for the encryption schemes used in securing cloud applications. While strengthening encryption schemes is a mandatory security practice, selecting an unsuitable encryption scheme poses a high-security risk to the application and subsequently degrades the performance of applications and business operations. Further research is needed to seek coherence on the strengths and shortcomings of various cloud encryption schemes to ascertain the scheme that is suitable for specific applications and maximize its functionality. This study focuses on the state-of-the-art encryption schemes employed in the cloud by conducting an in-depth review of current cloud-based encryption schemes. This study aims to serve as a reference to guide practitioners in selecting suitable encryptions for their cloud-based applications while maximizing functionality and keeping security in check.
Keyword head: cryptography; encryption; decryption; cloud; security

* Corresponding author: styleest2011@gmail.com

1. INTRODUCTION

The emergence of big data which co-evolved with cloud computing has made outsourcing data to cloud servers to be an absolute necessity for end users. The explosive growth in the volume, variety and velocity of data generated every second has made the cloud to be the de facto means of storing and managing applications [1-2]. Cloud computing brings several cutting-edge opportunities to the end users with a guaranteed unlimited amount of managed storage space, operational efficiency, collaborative platforms, productivity and a pervasive access to network infrastructures [3]. This positive side of cloud computing has driven large and resourceful enterprise such as Amazon Web Services [4], Dropbox [5], HP [6], Microsoft Azure [7] and others to embrace the cloud for their computations and subsequently this has set the cloud as an essential tool for everyday use [8].

Security concerns, such as privacy, unease over government inspection of data, authorization, inadequate access control, verification, back-door/trapdoor infusion into encryption algorithms, poor encryption and implementation with loopholes are the main challenges in Cloud Computing [9-12]. Such security concerns have driven an increasing use of state-of-the-art cryptographic techniques for ensuring the security of data in the cloud [13-19]. Encrypting user's data to curtail malicious attacks and providing security may address some of the security challenges in the cloud but there is need to use suitable encryption scheme for specific application to maximize the application functionality and not impede security and performance of the cloud service. Choosing a suitable encryption scheme for an application that allows certain functionality is often a huge challenge in cloud encryption.

Current novel cryptographic schemes such as Order-Preserving Encryption (OPE), Format-Preserving Encryption (FPE), Searchable Encryption (SE), Homomorphic Encryption (HE), Attribute-based Encryption (ABE) schemes may be suitable for addressing the privacy concerns of cloud computations. However, each scheme has a specific use case, for example, Searchable encryption allows enterprises to conduct searching of encrypted data such as keyword search, interval search, subset search, etc., as well as securely recovering private data, in this measure, guaranteeing the information security of un-trusted service providers [20].

Furthermore, the absence of use case of application-to-encryption standardization in cloud-based encryption platform implies a corresponding lack of clarity in the service offered by various cloud vendors/providers [21-22]. Cloud vendors are quick to guarantee that their service offers the best encryption suite. However, they fail to define the trade-offs between security and application functionality, and the critical effect that choosing the wrong encryption scheme may present on the client's data and operations [23-25].

This study aims to throw light on how to achieve a balance between security and application functionality for the specific use case of cloud encryption schemes. It draws a boundary between what a user will achieve in terms of performance when certain encryption algorithms are employed on cloud applications. This will help the user

understand the functional and security requirement of their cloud applications and the best security notions they can get when they employ certain encryption schemes. Moreover, this study will serve as a quick reference to practitioners in selecting specific encryption algorithms which are suitable for their cloud-based applications while maximizing their functionality for optimal performance. To this end, state-of-the-art cloud-encryption schemes are technically reviewed and compared in terms of the functionality of their algorithmic design, the security they offer and their suitability for specific use in the cloud. This study focuses explicitly on symmetric cryptographic algorithms where the client exclusively holds the key. In addition, we omit an overview of cloud systems as it is beyond the scope of this study. However, the encryption schemes discussed are employed in securing cloud-based applications.

This study is motivated by our observation that research exploring cloud-based encryption schemes from functionality versus use case perspectives is understudied. Therefore, the major contribution of this research is to advance the understanding of the adverse effect faced in the industry when an incompatible encryption scheme is matched with the wrong application. In a nutshell, we achieve this contribution by:

i. studying and categorizing modern symmetric-based encryption scheme.
ii. analyzing the security and suitability of specific encryption algorithms for cloud applications.
iii. suggesting practical use case of specific encryption algorithms that allow certain functionality while maximizing performance and providing optimal security.

The rest of this paper is organized as follows. Section 2 presents an overview of cryptographic schemes and the subsections details several standard symmetric encryptions schemes. Section 3 presents a tabular summary of different cloud-encryption algorithms, their advantages, limitations and specific use case. Finally, Section 4 concludes the study.

2. OVERVIEW OF CRYPTOGRAPHIC SCHEMES

The omnipresence of cryptography and its continued application to diverse fields in our daily lives makes research in this area evergreen. Modern cryptography lies at the heart of numerous processes and applications that incorporate Electronic Commerce, Internet Shopping, Online Gaming, GPS Navigation, Internet Banking, Business and Social Networks, Electronic Cars, Smart Systems and so forth in our information-aware society. Digital technology has changed the way in which we manage finances, procure goods, access healthcare and conduct research. People have become increasingly reliant on using technology to gain instant access to information, business collaboration, customers and family [26].

This trend of societal reliance on modern infrastructure has created an avenue for constant security threats and risk. This challenge has been demonstrated by incessant attacks on network servers, malware assaults, credit card fraud, botnet threats, hacking of banking applications, cell phones and computer attacks, security infringement,

phishing et cetera. To support our digital-dependent society for the present and future generation, and leverage on the opportunities that these technologies proffer, there is a need for trustworthy information infrastructure with strong security requirements. An essential building block to achieve the information security is cryptography [27-33].

Cryptography is the study of the art and science of hiding information to prevent a distrusted party from learning the original content of the message. Modern cryptography is mainly divided into two categories: symmetric-key cryptography and asymmetric cryptography. In the former, Alice and Bob share a key, which they do not disclose to an uninvited third party. Alice uses this key and an encryption algorithm called a cipher to transform the message from a plaintext to an unreadable form referred to as ciphertext. On receiving the ciphertext, Bob applies a decryption algorithm (reverse process of the encryption) and the shared key to recover the plaintext. In asymmetric cryptography, Alice and Bob, each has a pair of public key and private key.

The public key of Bob is used by Alice to encrypt her plaintext. The resultant ciphertext is decrypted by Bob using his pair of private key. As stated in the introduction section, this research is dedicated to symmetric-based cryptography. Readers interested in learning the functionality of modern asymmetric scheme are referred to [34-37]. A simplified scenario showing a classical symmetric mode of transmission is depicted in Figure 1.

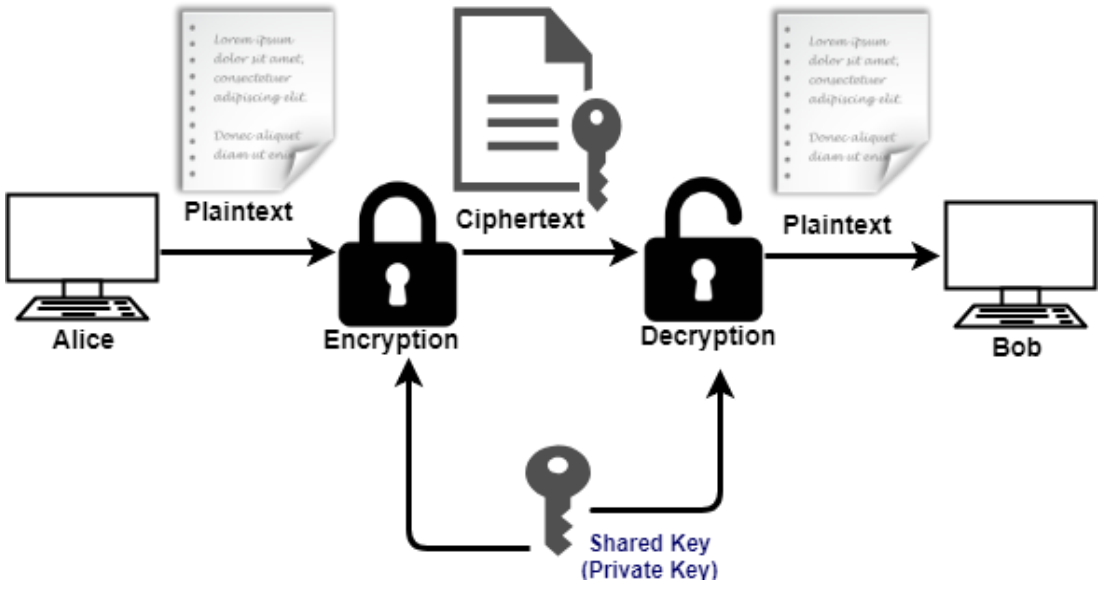

Figure 1: A Simplified Model of Symmetric Cryptosystem

Alice and Bob who wish to communicate secretly first agree upon a key K. For Alice to send a message/plaintext P to Bob, she encrypts P as an input in an encryption function Enc() with the key K to produce a ciphertext C to be sent to Bob.

$$C \leftarrow enc(P, K)$$

Bob recovers the plaintext, P by performing a decryption operation on the function Dec() which takes as input the values C and K.

$$P \leftarrow dec(C, K)$$

There is no single encryption scheme that offers complete security and functionality in one full bundle. Security and functionality are dependably inconsistent with each other. Therefore, some features of security must be traded at the expense of high functionality and vice versa. Understanding how to achieve this balance between security and functionality requires a detailed understanding of the encryption schemes. It is often a big challenge to select the right encryption scheme that supports specific applications and maximize their functionality. Consequently, it is required for the client to understand the security and functional requirements of their organization in tandem with the best level of security that a specific encryption scheme offers. This will guide him/her in choosing the specific algorithm that balances and augments all goals of performance and security for their application.

The next section explains in details standard cryptographic schemes which have been vetted to achieve completeness and have been declared to achieve provable security by internationally recognized standard bodies such as National Institute of Standards (NIST), American National Standards Institute (ANSI), et cetera. The features of each category of the algorithm are explained with examples to guide the practitioner in selecting the appropriate schemes for their cloud applications.

A. Symmetric Encryption Algorithm

In this section, we provide a tabular overview of some widely used, standard symmetric ciphers. We proceed to encapsulate the current state of the field by studying several standard symmetric-key ciphers, their strength and cryptanalytic attack that have been proposed to relegate their performance and use in the industry.

The first modern cipher which was proven secured (during the 70's) and generally accepted in the security community was Data Encryption Standard (DES) [38]. It is a Feistel block cipher that uses a 56-bit secret key to operate on a 64-bit block of data.

As at then, it was certified to be the best cipher after it has been subjected to rigorous and thorough scrutiny and deemed to have passed and achieved provable security. However, it was later proven to be insecure based on some successful attacks such as the differential attack, linear attack and also based on its small key size [39-41].

It key-size problem was addressed by proposing the triple DES (3DES), which employs a 168-bit secret key (56-bit keys are used three times to encrypt/decrypt the plaintext). Notwithstanding, it was discovered that other strategic attacks such as meet-in-the-middle attack, key-recovery attack and so forth were still successful in the 3DES. Subsequently, the Advanced Encryption Standard (AES) was presented as the successor of DES, 3DES and its variants. AES utilizes a fixed 128-bit block and unlike the DES and 3DES that uses a Feistel network, it employs a substitution-permutation network (SP-Network). It is fast both in hardware and software, requires a little memory and easy to implement [42].

In summary, AES is the de facto universal standard today as it has proven to be full-proof against most strategic cryptanalytic attack (except brute-force attack). Implementing AES alone cannot provide the full bundle of security needed to secure a cloud-application and as such, it is often used alongside other cryptographic primitives. It also has different mode of operation. Choosing a wrong mode of operation can degrade performance. A detailed treatment of state-of-the-art mode of operation is beyond the scope of this thesis. However, comprehensive details can be found in [43-45]. Table 1, presents a summary of the discussed encryption algorithms.

Table 1: Brief tabular summary of the evolution of symmetric encryption algorithms

Encryption Aspect	Key Size (bits)	Rounds	Cryptanalysis	Block Size (bits)	Time to crack	Input Size (Kilobyte)	Results of Encryption	Average Time	Encryption Speed
Data Encryption Standard (DES)	56-bit key	16 Rounds (Substitution, permutation, subkey)	Differential Cryptanalysis[39], Linear Cryptanalysis [40]	64-bit block	$7.2 * 10^{16}$	46, 104, 328, 905, 5202	27, 45, 75, 257, 987	310.2	5.06
Triple Data Encryption Standard (3DES)	112\168	48 Rounds	Meet in the Middle Attack[46], Key Recovery Attack[47],Related Key Attack[41]	64-bit block	$1.01 * 10^{18}$	46, 104, 328, 905, 5202	54, 87, 157, 270, 1108	310.2	5.06
Advanced Encryption Standard (AES)	128/192/256	10/12/14 Rounds (SubBytes, ShiftRows, MixColumns and AddRoundKey)	Brute-force	128-bit block	$3.4 * 10^{38}$/ $6.2 * 10^{57}$/ $1.1 * 10^{77}$	46, 104, 328, 905, 5202	55, 92, 167, 267, 1200	378.9	4.14
Blow Fish	32-448	16 Rounds	Birthday attack like http	64-bit	$1.01 * 10^{18}$	46, 104, 328, 905, 5202	40, 47, 87, 123, 157	96.3	16.32
RC2	40-1024	18 Rounds	Related Key	64-bit	$1.01* 10^{18}$	46, 104, 328, 905, 5202	56, 92, 176, 306, 1507	489.6	3.21
RC5	128/192/256	12 Rounds	Differential attack	64/128-bit	$3.4 * 10^{38}$/ $6.2 * 10^{57}$/ $1.1 * 10^{77}$	46, 104, 328, 905, 5202	42, 63, 110, 152, 765	247.1	6.36

3. CLASSIFICATION OF CLOUD ENCRYPTION SCHEMES

To achieve a reasonable level of security in the cloud, encryption schemes are often combined with some cryptographic primitives to optimize performance. For instance, a encryption scheme may provide integrity but fail to provide authentication and so forth. Hence, the need for combining certain functions. The combination of these algorithms forms the structure of the encryption. We discuss this structural classification of cloud encryption schemes broadly in this section.

A. Regular Encryption Scheme

Regular encryption is unstructured encryption schemes that provide data confidentiality. They can be used to encrypt data-type which contains arbitrary features such as Text Documents, Presentations, PDF, Spreadsheet. Regular encryption provides exceptionally strong security guarantees and as such allows confidential data to be hidden and secured under reasonable assumptions. This means that an attacker that intercepts an encrypted cloud data at rest or transit cannot determine the key or any bit or function of the message. A complete suite of Regular encryption provides integrity and authenticity by making it impossible for an attacker to modify the ciphertext without the receiver noticing. The length of the ciphertext and plaintext may or may not be equal and as such, a third party cannot tell if two ciphertexts correspond to equal messages.

The solid security notions of regular encryption scheme impact application functionality. Functions such as search, preview of documents, mathematical and logical operation and so forth are impeded when regular encryptions are employed. Therefore, regular encryption schemes are suitable for data in storage as many functions are not required in this state. Regular encryption should be used for highly confidential data that requires fewer functions.

A generic example of how regular encryption works is shown below:

Plaintext: CRYPTOGRAPHY IS INTERESTING

Ciphertext: $Dfh79!mf +96E@99j2!#$^*(*)&7)]P

Examples of regular encryption scheme are Data Encryption Standard (DES), Blowfish, Advanced Encryption Standard (AES). To achieve data confidentiality, data integrity and user authentication using a regular encryption scheme, state-of-the-art encryption suite must be employed such as using Advanced Encryption Standard with a suitable mode of operation. It is crucial to use the correct mode to achieve sufficient performance. Example of such mode includes Advanced Encryption Standard-Galois Counter Mode (AES-GCM), Advanced Encryption Standard-Cipher Block Chain (AES-CBC), Advanced Encryption Standard-Electronic Code Book (AES-ECB),

Advanced Encryption Standard-Cipher Feedback (AES-CFB), Advanced Encryption Standard-Output Feedback (AES-OFB), and so forth [45,48-49].

B. Format Preserving Encryption (FPE)

In Format-Preserving Encryption (FPE), a plaintext is encrypted into the same format and type as the ciphertext. FPE can be used to encrypt confidential data types such as Payment details (credit/debit card), Phone Numbers, Social Security Numbers (SSN), Personal Identifiable Information (PII) which are often used in healthcare, financial databases and applications.

FPE preserves data format for consistent improvement and migration of legacy application to the cloud. This implies that the ciphertext retains the character and length of the plaintext. For instance, a 10-digit phone number would be encrypted into a 10-digit phone number, a valid 16-digit credit card number would be encrypted into a valid 16-digit phone number, an English word would be encrypted into an equal length English word. In addition, FPE allows certain functionality to be enjoyed in the cloud, for instance: a subset of a data item can be kept in the clear, allowing applications to use the data without decryption.

As an example, the last four digits of a credit card may be retained, this allows the use of the data without any need to decrypt first. In addition, FPE provides referential integrity such that a data item encrypted twice will result in the same ciphertext. This feature allows analytic applications that employ confidential data for database keys or item counts to directly use the encrypted data item. By implementing FPE, many applications can run with encrypted versions of confidential data with minimal or no changes to the existing business processes or system architecture, thereby reducing cost.

In 2013, the National Institute of Standards and Technology (NIST) recommended three modes of operation for FPE: the FF1, FF2 and FF3 mode. In each of these three modes of operation, the Advanced Encryption Standard (AES) is used to construct a round function within the Feistel structure for encrypting data. FF1 and FF2 mode are been reevaluated as there are security concerns for their longevity. Another mode of FPE suggested which complies with security requirement and regulation is the SP 800-38G which is fully validated with FIPS 140-2 by NIST [50].

However, it is very important to state here that in April 2017, the NIST discovered a cryptanalytic attack on the FF3 mode for FPE and as such was declared unsuitable for general and technical use [51-54]. With this discovery, the FF3 mode of the FPE which was gaining widespread use by large organizations was halted and currently, the cryptographic community is seeking alternative secure modes of implementing FPE. A complete description of the FPE can be found here [55].

A generic example of how FPE works is shown below:

Plaintext: 0133-234-9187

Ciphertext: 0142-756-1092

The premise of building a secure scheme is to make the encryption as unpredictable as possible. For instance, given a ciphertext, an attacker should be unable to deduce any characteristics of the plaintext. However, FPE fails to achieve this "unpredictability basis" as it inherently reveals the format, character or type of the plaintext. In addition, FPE is specific for short-bit messages such as credit-card, social security numbers, dates [56-57]. FPE is difficult to be generalized. For instance, applying FPE to large clear text values such as encrypting an email or large documents to be stored in the cloud requires each word to be encrypted into another word of the same length and the same language, consequently, impacting time.

In summary, a regular encryption scheme, for instance, an AES-CBC mode that is used to encrypt a 9-digit Social Security Number may be encrypted as %F1+67eh#Vb/--lz3 which is longer than 9 characters and composed of a character type which is no longer digit will surely impact functionality in a complex legacy environment where the application expects to get only a 9-digit value. FPE tackles this problem adequately by encrypting in the same format and type. However, it leaks equality between the encrypted data and the original data. Therefore, it is advisable to use FPE scheme when the security requirements of your application can tolerate equality leakage and if the application requires a server-side input validation.

C. Searchable Encryption Scheme

Searchable encryption allows a party to outsource the storage of his data to another party in a private manner while maintaining the ability to selectively search over it. Searchable encryption scheme can be built using either a word-by-word approach or an index/keyword-based approach. In the word-by-word scheme, each word is encrypted independently with searchable encryption. This makes it possible for searching specific words in the document. This provides the functionality to search any words in the file.

However, it takes a long search time for a large number of the document set. In the keyword-based scheme, keywords are extracted from the encrypted document and preserved. The keywords are encrypted and inserted in a metadata header. This allows a fast search operation over a large document. However, updating and storing the index can create a large overhead.

A generic example of how Searchable encryption works is shown below:

Plaintext: 1:26784950160
2:74905762561
3:76387889494

Ciphertext: 1:SdlpwnyPRmmTkllgdfb=
2: wnyPRmgdfMMlqoppa=
3: TkllgdfbSdlpwnqwPlMeh=

D. Order Preserving Encryption (OPE)

In Order Preserving Encryption (OPE), the encryption function preserve the numerical ordering of the plaintexts. This implies that the plaintext is encrypted into the same order as the ciphertext. OPE can be used to encrypt confidential data with values such as numeric and alpha-numeric types.

OPEs comes with a lot of functionality. It allows efficient range queries on encrypted data. This implies that a remote untrusted database server can index the confidential data it receives, in an encrypted form and in a data structure that allows efficient range queries. For instance, a request can be made for the server to return ciphertexts in the database whose decryptions fall within a specific range, say [x,y].

OPEs allows queries to be processed, indexing, comparison and sorting on the encrypted data. In addition, it allows a standard database to be incorporated into the current database without any need to modify the framework. For instance, a new value can be added to a column or an existing value updated without any alteration to other values. OPEs can be securely implemented in environments where the adversary can gain access to the encrypted database yet have no prior domain information.

A generic example of how OPE works is shown below:

Plaintext: 1:267849501
2:7490576256
3:76387889494

Ciphertext: 1:758939301
2:2314562778
3:47482909187

The major drawback of OPE is that it leaks the order of the plaintext which invariably implies that related information would be revealed as well. For instance; an adversary can determine the relative distance between the encrypted plaintext. A short length

ciphertext is probably going to correspond to a short length plaintext. In addition, OPEs are susceptible to tight estimation exposure. This means an attacker can easily guess the approximate value of the plaintext based on the ciphertext. For instance, an adversary can guess a 6-digit Chinese postal code and estimate if it lies within an interval with a high probability.

E. Selective Encryption Schemes

Selective encryption is an encryption scheme where a subset of the message is encrypted. It saves computational complexity by encrypting non-compliant substrings of a portion of the data while still preserving a substantial level of security. They are mostly used to encrypt videos, audios and images.

In selective encryption, the data is divided into two parts. The first part is unencrypted, and it is made public and accessible to all users. The second part is encrypted and is available to only authorized users. The encrypted part is usually a small portion as it limits functionality while the larger part which is unencrypted allows several functionalities. Most organizations rely on access control system to protect their digital contents. For instance, mobile phones, PDA and other versatile terminals are employed for transmitting multimedia contents (video, image, voice) while still requiring copyright protection and access control. Video content has low-security requirement, hence, instead of encrypting the complete video, the content quality can be degraded to compel people into buying a full-quality content of the original video [58,59,60]. This is where selective encryption comes into play. Nevertheless, security of information and encryption technique can be relatively compared to security of human and properties using various method such as DNA [59], big data [60, 61] and other approaches [62, 63, 64] for performance.

The major challenge with selective encryption is that data is regularly transferred at a fast pace, and a user may accidentally post information that may abuse compliance regulations.
A generic example of how Selective Encryption works is shown below:

Plaintext: My credit card number is 5120-0142-7056-1092

Ciphertext: My credit card number is ofjf#gp4TL8+=96E@] Q

Selective encryption should be used when the application needs few or no functional operation (such as sorting, search) and there is a need to protect confidential policy-based and regulatory compliant data. A good way of achieving optimal security using

selective encryption is to use AES with an adequate complementary mode such AES-GCM.

4. TABULAR SUMMARY OF SYMMETRIC CLOUD ENCRYPTION SCHEMES AND THEIR USE CASE

A summary of the advantages, limitations and use case of cloud-encryption schemes is shown in Table 2.

Table 2: Summary of symmetric-structured cloud-encryption schemes

Encryption Scheme	Advantages	Functional Deficiency	Use Case
Regular or Regular Encryption Scheme. Example: Advanced Encryption Standard (AES), Data Encryption Standard (DES)	✓ Provides data confidentiality. A well-crafted regular encryption scheme such as AES-GCM implies that an adversary cannot recover the key or any chunk of the message given the ciphertext. ✓ Provides provable security guarantee as confidential data are completely hidden under reasonable assumptions. ✓ Good encryption scheme for remote storage.	✓ Features such as searching, mathematical and logical operation, document preview are impeded with a regular encryption.	Regular encryption should be used only if securing the application is more important than usability as certain features become unusable if employed.
Format Preserving Encryption (FPE)	✓ Preserves the Format of the Plaintext. ✓ Preserves the length of the Plaintext. ✓ A well-crafted FPE provides random-looking ciphertext for distinct messages. ✓ It allows the ciphertext to be used and stored in the same way as the plaintext and hence there is no compelling need to change the structure of the database table. ✓ It guarantees data privacy by preserving format-specific properties during encryption while other parts of the message remain hidden. ✓ The application of FPE in Electronic healthcare systems allows secure matching and sharing of patient records in the hospital database. ✓ FPE allows anonymity of personal data, such as credit card details, et cetera ✓ FPE permits revamping of database security in a way which is transparent to several applications and minimally intrusive to others. ✓ Partial encryption can be executed which effectively allows search operations over a large set of encrypted numbers such as credit card.	✓ An attacker can easily determine the length of the plaintext from the ciphertext intercepted. He knows the format and type of data he is targeting, for instance, a credit card number, a social security number. ✓ Size restriction: An input of K-digit number must produce an output of K-digit number. In hindsight, FPE cannot be expanded. ✓ Plaintext size constraint: effective only for small-length plaintext. For instance, credit cards are 58 bits long, SSN contains about 28 bits. ✓ Does not provide authenticity of the sender. ✓ Does not provide data integrity as an attacker can intercept the message and modify it with random bits of equal format, type and length, after which he sends the modified message without the receiver noticing any changes.	FPE should be used if a distinct format is required by an application. It should be used if the security requirements of an application can tolerate the leakage of some features of the plaintext. Also, it should be employed if the application requires server-side input validation checking.

Encryption Scheme	Advantages	Functional Deficiency	Use Case
Searchable Encryption Scheme	✓ Saves time as the user need not download complete data before searching. ✓ Multiparty search between ciphertext without access to plaintext. ✓ Gives response based on the server's request without revealing the content of the client's query to the server. ✓ Symmetric searchable encryption enables only the secret key holders to generate ciphertexts and to produce trapdoors for searching through the texts.	✓ Keyword search leaks equality of the keyword and this makes statistical attack possible.	Searchable encryption allows efficient search on an encrypted document and should be used when the user is willing to trade some security features in order to search through the encrypted data effectively.
Order Preserving Encryption (OPE)	✓ Decryption preserves the order of the encrypted data. ✓ Allows efficient range query and checking, ranking on the encrypted data. ✓ Allows high-level user functionality such as sorting, searching, indexing. ✓ A well-crafted OPE hides everything else except the order of the plaintext. ✓ Accurate query results are achieved. No false hit	✓ The order of the plaintext is leaked, subsequently leaking related information. ✓ Not a fully adopted or widely used encryption scheme for cloud applications, therefore, using it might present some risks that have not been detected over time and which may be exploited by an adversary. ✓ OPEs are susceptible to tight estimation exposure as an attacker can select any number of encrypted values and decrypt them into their corresponding unencrypted values. In retrospect, an adversary can learn a large portion of the bits of an underlying plaintext given the ciphertext.	OPE should be used when indexing, sorting and searching of data outweigh the requirement of security. In hindsight, it should be used when the user is okay with trading some security features to enjoy some usability features.
Selective Encryption Scheme	✓ It's capability to encrypt delicate data to guarantee regulatory compliance while leaving other data unencrypted helps to preserve much of the application functionality. ✓ It saves computational power, time and overhead cost. ✓ Encryption is fast. As only a selected bitstream is encrypted.	✓ Data is regularly transferred at a fast pace, and a user may accidentally post information that may abuse compliance mandates.	Selective encryption should be used on policy-based and solid configurable data which requires no sorting or searching.

5. CONCLUSION

Cloud computing offers numerous advantages to users, including substantial storage capacity and high computation power. However, concerns around security, application performance, privileged access to data, et cetera, raise trust and privacy concerns. Several encryption algorithms have been proposed to address this security concerns of cloud applications. Nevertheless, most of this encryption schemes degrade security and impedes application functionality when a wrong encryption scheme is paired with an unsuitable application. Moreover, users are disappointed when they require certain functionality from their applications and it seems impossible due to the kind of encryption they selected for their business. Previous studies have shown that there is no single encryption algorithm that is ideal for all situations. Hence, it is critical to understand where encryption can help secure data and where it limits functionality and break applications.

Achieving sufficient functionality and optimal security requires an in-depth knowledge of the security and functional requirement of the business coupled with an understanding of the algorithmic design of the encryption scheme to be used. For optimal performance, the requirements must be understood to achieve a balance between application functionality and security. To this end, we presented a comprehensive review of state-of-the-art symmetric encryption schemes employed in cloud security. It is our goal that this study will act as a manual to guide practitioners in choosing the correct encryption scheme for their applications.

Notwithstanding, we recommend that international bodies (such as NIST, ISO) provide a standard manual which each cloud vendor/service providers must supply to their users stating specific use case of encryption for each application. The manual should clearly illuminate the pros and cons of using a specific encryption scheme and also clarify the balance between security and application functionality to prospective users/practitioners. Standardization will compel cloud service providers and vendors to come up with concrete parameter sets for specific use cases of each encryption scheme for specific cloud-hosted applications. Moreover, competition between different service providers to give the best options to their clients will foster extensive research towards this path until a balance between maximal security versus application functionality is reached.

REFERENCES

1. Wu, X., Zhu, X., Wu, G. Q., & Ding, W. (2014). Data mining with big data. IEEE transactions on knowledge and data engineering, 26(1), 97-107.
2. Ranjan, R. (2014). Streaming big data processing in data center clouds. IEEE Cloud Computing, 1(1), 78-83.
3. Moataz, T., Justus, B., Ray, I., Cuppens-Boulahia, N., Cuppens, F., & Ray, I. (2014, July). Privacy-preserving multiple keyword searches on outsourced data in the clouds. In IFIP Annual Conference on Data and Applications Security and Privacy (pp. 66-81). Springer, Berlin, Heidelberg.
4. Amazon web services. https://aws.amazon.com. (accessed 15 August 2018).
5. Dropbox. http://www.dropbox.com. (accessed 15 August 2018).
6. Hp public cloud. http://www.hpcloud.com/console. (accessed 15 August 2018).
7. Microsoft Azure. http://azure.microsoft.com. (accessed 15 August 2018).
8. Vaquero, L. M. (2011). EduCloud: PaaS versus IaaS cloud usage for an advanced computer science course. IEEE Transactions on Education, 54(4), 590-598.
9. Popović, K., & Hocenski, Ž. (2010, May). Cloud computing security issues and challenges. In *MIPRO, 2010 proceedings of the 33rd international convention* (pp. 344-349). IEEE.
10. Li, S., Huang, L., Fu, A., & Yearwood, J. (2017). CExp: secure and verifiable outsourcing of composite modular exponentiation with single untrusted server. *Digital Communications and Networks*, *3*(4), 236-241.
11. Singh, S., Jeong, Y. S., & Park, J. H. (2016). A survey on cloud computing security: Issues, threats, and solutions. *Journal of Network and Computer Applications*, *75*, 200-222.
12. Li, R., Asaeda, H., Li, J., & Fu, X. (2017). A distributed authentication and authorization scheme for in-network big data sharing. *Digital Communications and Networks*, *3*(4), 226-235.
13. Alowolodu, O. D., Alese, B. K., Adetunmbi, A. O., Adewale, O. S., & Ogundele, O. S. (2013). Elliptic curve cryptography for securing cloud computing applications. *International Journal of Computer Applications*, *66*(23).
14. Jaber, A. N., & Zolkipli, M. F. B. (2013, November). Use of cryptography in cloud computing. In *Control System, Computing and Engineering (ICCSCE), 2013 IEEE International Conference on* (pp. 179-184). IEEE.
15. Matsuda, S., & Moriai, S. (2012, September). Lightweight cryptography for the cloud: exploit the power of bitslice implementation. In *International Workshop on Cryptographic Hardware and Embedded Systems* (pp. 408-425). Springer, Berlin, Heidelberg.
16. Stergiou, C., Psannis, K. E., Kim, B. G., & Gupta, B. (2018). Secure integration of IoT and cloud computing. *Future Generation Computer Systems*, *78*, 964-975.

17. López-Alt, A., Tromer, E., & Vaikuntanathan, V. (2012, May). On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing* (pp. 1219-1234). ACM.

18. Omolara, A. E., Jantan, A., Abiodun, O. I., & Poston, H. E. (2018). A Novel Approach for the Adaptation of Honey Encryption to Support Natural Language Message. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (Vol. 1).

19. Li, J., Zhang, Y., Chen, X., & Xiang, Y. (2018). Secure attribute-based data sharing for resource-limited users in cloud computing. *Computers & Security*, *72*, 1-12.

20. Fun, T. S., & Samsudin, A. (2016). A survey of homomorphic encryption for outsourced big data computation. *KSII Transactions on Internet and Information Systems (TIIS)*, *10*(8), 3826-3851.

21. Singh, A., & Chatterjee, K. (2017). Cloud security issues and challenges: A survey. *Journal of Network and Computer Applications*, *79*, 88-115.

22. Li, P., Li, J., Huang, Z., Gao, C. Z., Chen, W. B., & Chen, K. (2017). Privacy-preserving outsourced classification in cloud computing. *Cluster Computing*, 1-10.

23. Coppolino, L., D'Antonio, S., Mazzeo, G., & Romano, L. (2017). Cloud security: Emerging threats and current solutions. *Computers & Electrical Engineering*, *59*, 126-140.

24. De Carvalho, C. A. B., de Castro Andrade, R. M., de Castro, M. F., Coutinho, E. F., & Agoulmine, N. (2017). State of the art and challenges of security SLA for cloud computing. *Computers & Electrical Engineering*, *59*, 141-152.

25. Iqbal, S., Kiah, M. L. M., Dhaghighi, B., Hussain, M., Khan, S., Khan, M. K., & Choo, K. K. R. (2016). On cloud security attacks: A taxonomy and intrusion detection and prevention as a service. *Journal of Network and Computer Applications*, *74*, 98-120.

26. Mulholland, J., Mosca, M., & Braun, J. (2017). The day the cryptography dies. *IEEE Security & Privacy*, *15*(4), 14-21.

27. Wyseur, B. (2011). White-box cryptography. In *Encyclopedia of Cryptography and Security* (pp. 1386-1387). Springer, Boston, MA.

28. Schellekens, D., Wyseur, B., & Preneel, B. (2008). Remote attestation on legacy operating systems with trusted platform modules. *Science of Computer Programming*, *74*(1-2), 13-22.

29. Wyseur, B., Michiels, W., Gorissen, P., & Preneel, B. (2007, August). Cryptanalysis of white-box DES implementations with arbitrary external encodings. In *International Workshop on Selected Areas in Cryptography* (pp. 264-277). Springer, Berlin, Heidelberg.

30. Omolara, O. E., Oludare, A. I., & Abdulahi, S. E. (2014). Developing a modified Hybrid Caesar cipher and Vigenere cipher for secure Data Communication. *Computer Engineering and Intelligent Systems*, *5*, 34-46.

31. Omolara, A. E., Jantan, A., Abiodun, O. I., & Arshad, H. (2018). An Enhanced Practical Difficulty of One-Time Pad Algorithm Resolving the Key Management and Distribution Problem. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (Vol. 1).

32. Omolara, A. E., Jantan, A., Abiodun, O. I., & Poston, H. E. (2018). A Novel Approach for the Adaptation of Honey Encryption to Support Natural Language Message. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (Vol. 1).

33. Sfar, A. R., Natalizio, E., Challal, Y., & Chtourou, Z. (2018). A roadmap for security challenges in the Internet of Things. *Digital Communications and Networks*, *4*(2), 118-137.

34. Bellare, M., & Rogaway, P. (1994, May). Optimal asymmetric encryption. In *Workshop on the Theory and Application of-of Cryptographic Techniques* (pp. 92-111). Springer, Berlin, Heidelberg.

35. Bellare, M., Desai, A., Pointcheval, D., & Rogaway, P. (1998, August). Relations among notions of security for public-key encryption schemes. In *Annual International Cryptology Conference* (pp. 26-45). Springer, Berlin, Heidelberg.

36. Ma, M., He, D., Kumar, N., Choo, K. K. R., & Chen, J. (2018). Certificateless searchable public key encryption scheme for industrial internet of things. *IEEE Transactions on Industrial Informatics*, *14*(2), 759-767.

37. Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *21*(2), 120-126.

38. National Institute of Standards and Technology: Data Encryption Standard. FIPS publication 46-3 (1977). http://csrc.nist.gov/publications/ fips/fips46-3/fips46-3.pdf. 21

39. Biham, E., & Shamir, A. (1991). Differential cryptanalysis of DES-like cryptosystems. *Journal of CRYPTOLOGY*, *4*(1), 3-72.

40. Matsui, M. (1993, May). Linear cryptanalysis method for DES cipher. In *Workshop on the Theory and Application of-of Cryptographic Techniques* (pp. 386-397). Springer, Berlin, Heidelberg.

41. Phan, R. C. W. (2004, February). Related-key attacks on triple-DES and DESX variants. In *Cryptographers' Track at the RSA Conference* (pp. 15-24). Springer, Berlin, Heidelberg.

42. National Institute of Standards and Technology: Advanced encryption standard. FIPS publication 197 (2001). http://csrc.nist.gov/publications/fips/ fips197/fips-197.pdf. 23, 62.

43. Rogaway, P., Bellare, M., & Black, J. (2003). OCB: A block-cipher mode of operation for efficient authenticated encryption. *ACM Transactions on Information and System Security (TISSEC)*, *6*(3), 365-403.

44. Whiting, D., Housley, R., & Ferguson, N. (2003). *Counter with CBC-mac (ccm)* (No. RFC 3610).

45. McGrew, D. A., & Viega, J. (2004, December). The security and performance of the Galois/Counter Mode (GCM) of operation. In *International Conference on Cryptology in India* (pp. 343-355). Springer, Berlin, Heidelberg.

46. Waliullah, M., & Gan, D. (2014). Wireless LAN security threats & vulnerabilities. International Journal of Advanced Computer Science and Applications, 5(1).

47. Maged Hamada Ibrahim (2015). Anonymously Authenticated Transmission on the Cloud with Traceability. *International Journal of Advanced Computer Science and Applications, 6(9).*

48. Dworkin, M. J. (2007). *Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC* (No. Special Publication (NIST SP)-800-38D).

49. Dworkin, M. (2016). Recommendation for block cipher modes of operation: methods for format preserving encryption. *NIST Special Publication*, *800*, 38G.

50. Computer Security Resource Centre. April 2017. Recent Cryptanalysis of FF3. https://csrc.nist.gov/News/2017/Recent-Cryptanalysis-of-FF3. Retrieved 27th August 2018

51. Durak, F. B., & Vaudenay, S. (2017, August). Breaking the FF3 format-preserving encryption standard over small domains. In *Annual International Cryptology Conference* (pp. 679-707). Springer, Cham.

52. Bellare, M., Hoang, V. T., & Tessaro, S. (2016, October). Message-recovery attacks on Feistel-based format preserving encryption. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 444-455). ACM.

53. Betül Durak, F., & Vaudenay, S. (2017). *Breaking the ff3 format-preserving encryption standard over small domains*. Cryptology ePrint Archive, Report 2017/521.

54. Bellare, M., Rogaway, P., & Spies, T. (2010). The FFX mode of operation for format-preserving encryption. *NIST submission*, *20*.

55. Hoover, D. N. (2015). *U.S. Patent No. 8,948,376*. Washington, DC: U.S. Patent and Trademark Office.

56. Bower, M. F., Pauker, M. J., & Spies, T. (2012). *U.S. Patent Application No. 13/155,156*.

57. Bellare, M., Ristenpart, T., Rogaway, P., & Stegers, T. (2009, August). Format-preserving encryption. In *International Workshop on Selected Areas in Cryptography* (pp. 295-312). Springer, Berlin, Heidelberg.

58. Massoudi, A., Lefebvre, F., De Vleeschouwer, C., Macq, B., & Quisquater, J. J. (2008). Overview of selective encryption of image and video: challenges and perspectives. *Eurasip Journal on information security*, *2008*, 5.

59. A. I. Oludare, A. Jantan, A. E. Omolara, M. M. Singh, M. Anbar, Z. F. Zaaba "Forensic DNA profiling for identifying an individual crime" International Journal of Civil Engineering and Technology (IJCIET), July, 2018, PP. 755-765.

60. A. E. Omolara, A. Jantan, O. I. Abiodun, M. M. Singh, M. Anbar, D. V. Kemi, State-of-the-art in big data application techniques to financial crime: a survey. International Journal of Computer Science and Network Security, 2018, 18(7), 6-16.

61. A. I. Oludare, A. Jantan, E. O. Abiodun, M. M. Singh, Z. L. Abubakar, A. M. Umar, Big data: an approach for detecting terrorist activities with people's profiling," Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong, Vol I IMECS 2018, 14-16, March 2018.

62. L. S. Choon, A. Samsudin, R. Budiarto, Lightweight and cost-effective MPEG video encryption. In Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. 2004 International Conference on(pp. 525-526). April 2004, IEEE.

63. A. I. Oludare, A. Jantan, A. E. Omolara, M. M. Singh, A. Mohammed, D. V. Kemi, Terrorism prevention: a mathematical model for assessing individuals with profiling, International Journal of Computer Science and Network Security, July 2018, vol.10, no.12.

64. Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. Heliyon, 4(11), e00938.