Asian Journal of Information Technology 13 (9): 494-505, 2014
ISSN: 1682-3915

A Survey on Data Encryption Tecniques in Cloud Computing

S. Balasubramaniam and V. Kavitha
Department of Computer Science Engineering,
University College of Engineering, 629004 Nagercoil,
Anna University, Chennai, Tamil Nadu, India

Abstract: Cloud computing comes into focus only when you think about what IT always needs: a way to increase capacity or add capabilities on the fly without investing in new infrastructure, training new personnel or licensing new software. Cloud computing encompasses any subscription-based or pay-per-use service that in real time over the Internet, extends IT's existing capabilities. Cloud computing is an updated version of utility computing: basically virtual servers available over the internet or anything you consume outside the firewall is "in the cloud" including conventional outsourcing. There are nine primary reasons why mid-sized companies (100-500 employees) are adopting Cloud services such as defer and avoid expansion of data center facilities, strengthen Business continuity, prepare for disaster recovery, earn a high return on investment, improve security compliance, defer server purchases and upgrades, shift from capital to operating budget, strive to go green, decrease IT staff. Cloud computing has several major issues and concerns such as data security, trust, regulations and performance issues. Security is one of the most critical aspects in cloud computing. There are various risks associated with security but one of the major issues is the security during data transmission and data storage. Before outsourcing the data into cloud servers, data's are encrypted using Encryption algorithms by the users for security purpose. There are various Encryption algorithms are used to encrypt the data. This study concentrated on various encryption techniques are discussed for providing solutions to cloud security.

Key words: Encryption, decryption, DES, AES, RSA

INTRODUCTION

Cloud is a new name for services such as webmail that have been around for nearly a decade. Cloud computing involves programs or services that run on Internet servers. Today more and more services like documentation, storage, workflows, email and office applications like accounting, HR, purchase, CRM among others are being delivered form the cloud. The National Institute of Standards and Technology (NIST) in US proposed three cloud computing service models: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a service (IaaS) (NIST). Cloud computing holds the promise of providing computing as a fifth utility after other four utilities such as water, gas, electricity and telephone (Buyya *et al*., 2013). There are several reasons for adopting cloud computing but on demand scalability, reduction in capital expenditure and reduction in operations expenditure were top three reasons selected by many organizations. Moving from a highly secure data centre to internet based cloud model will require great emphasis on security and privacy. Migration to cloud model has implications for all types of services and applications to be hosted on cloud and may require major modifications or enhancements in existing processes. Data security and privacy in cloud computing are engaging the attention of user organizations and cloud service providers namely, privacy protection, data security and Complaint issues are wide open, since the data is no longer under the control of owner.

Encryption is essential to cloud computing. Encryption is one of the most effective data protection controls available today. Encryption integrity is based on the technologies and processes governing the cryptographic security services. Encryption is a primary data (and application) protection technique. Encryption solutions implement cryptosystems that utilize one or more cryptography algorithms. Very often, these solutions combine asymmetric and symmetric cryptography where asymmetric keys are used to set up symmetric keys on both ends of a communication path and then the symmetric keys are used for content

Corresponding Author: S. Balasubramaniam, Department of Computer Science Engineering, University College of Engineering, 629004 Nagercoil, Anna University, Chennai, Tamil Nadu, India

encryption. Different encryption algorithms have different strengths. The emergence of cloud delivery of security services now means that encryption capabilities can not only be used to secure data in the cloud but can also be offered through the cloud to enable organizations of all kinds to more easily protect sensitive data (https://cloudsecurityalliance.org/download/secaas-category-8-encryption-implementation-guidance/). Many encryption techniques are existing which are used to avoid the Data Leakage. Different encryption techniques are used to protect the confidential data from unauthorized use. Everyday new methods of encryption techniques are discovered. Encryption is the process of converting Plain Text into Cipher Text. Cryptography uses the encryption technique to send confidential messages through an insecure channel. The process of encryption requires two things an Encryption algorithm and a key. An Encryption algorithm means the technique that has been used in encryption. Encryption takes place at the sender side.

ENCRYPTION ALGORITHMS

There are different security algorithms are available to eliminate the concerns regarding data security and privacy while accessing applications on cloud. Some of the algorithms are DES, AES, Blowfish are symmetric key algorithms, in which a single key is used for both encryption/decryption of messages. RSA is Public Key algorithm also called as Asymmetric Key algorithm, the algorithm that uses different keys for encryption and decryption purposes (Table 1).

DES: Data Encryption Standard (DES) was developed in early 1970's by IBM. The key length of DES algorithm is 56 bits. The Data Encryption Standard (DES) is a block cipher. It encrypts data in blocks of size 64 bits each. That is 64 bits of plain text goes as input to DES which produces 64 bits of cipher text. The same algorithm and key are used for encryption and decryption, with minor differences. The key length of this algorithm is 56 bits; however a 64 bits key is actually input. Therefore, DES is a Symmetric Key algorithm (Stallings, 2003).

AES: Advanced Encryption Standard (AES) was designed by NIST in 2001. The key length of AES algorithm is 128, 192, 256 bits. AES is a symmetric block cipher. AES Symmetric Key Encryption algorithm is used with key length of 128 bits. AES is used widely now a day for security of cloud. When migration of data to the chosen Cloud Service Provider (CSP) happens and in future whenever an application uploads any data on cloud, the data will first encrypted using AES algorithm and then sent to the CSP (Stallings, 2003). Once encrypted, data is uploaded on the cloud, any request to read the data will occur after it is decrypted on the users end and then plain text data can be read by user. The plain text data is never written anywhere on cloud. This encryption solution is transparent to the application and can be integrated quickly and easily without any changes to application (Stallings, 2003). The key is never stored next to the encrypted data because it may compromise the key also. To store the keys, a physical key management server can be installed in the user's premises. This encryption protects data and keys and guarantees that they remain user's control and will never be exposed in storage or in transit. AES has replaced the DES as approved standard for a wide range of applications. The encryption consists of 10 rounds of processing for 128 bit keys, 12 rounds for 192 bit keys and 14 rounds for 256 bit keys (Khanna and Jaiswal, 2013). The four rounds are called SubBytes, ShiftRows, MixColumns and AddRoundKey.

Blowfish: Blowfish was designed by Bruce Schneider in 1993. The key length of Blowfish algorithm is 128-448 bits. Blowfish is a Symmetric Key Cryptographic algorithm. Blowfish encrypts 64 bit blocks with a variable length key of 128-448 bits. Blowfish suits applications where the key remains constant for a long time but not where the key changes frequently (Stallings, 2003). Blowfish is a symmetric block Cipher Encryption algorithm meaning that it uses the same secret key to both encrypt and decrypt messages and divides a message up into fixed length blocks during encryption and decryption. The block length for this algorithm is 64 bits; messages that aren't a multiple of eight bytes in size must be padded. It is faster than most Encryption algorithms when implemented on 32 bit microprocessors with large data

Table 1: Characteristics of Encryption algorithms

Characteristics	DES	AES	Blowfish	RSA
Initial vector size	64 bits	128 bits	64 bits	1024 bits
Key size	56 bits	128, 192, 256 bits	32-448 bits	1024 bits
Key used				
Encrypt/decrypt	Same key	Same	Same	Public
Data encryption on capacity	Less than AES	Used for encryption of large amount of data	Less than AES	Used for encryption of small data
Security	Security applied to both providers and user	Secure for both providers and user	Security applied to both providers and user/client side	Secure for user only
Execution time	Equals to AES	Faster than others	Lesser time to execute	Requires maximum time

caches (Khanna and Jaiswal, 2013). It takes a variable length key from 32-448 bits and making it ideal for securing data. The application of this algorithm is communications link or an automatic file encrypter where the key does not change often (Khanna and Jaiswal, 2013).

RSA: RSA is a Public Key algorithm invented by Rivest, Shamir and Adlemen in 1978 and also called as Asymmetric Key algorithm. The algorithm that uses different keys for encryption/decryption purposes. The key size of RSA algorithm is 1024 bits. RSA is basically an Asymmetric Encryption/Decryption algorithm. It is asymmetric in the sense that here public key distributed to all through which one can encrypt the message and private key which is used for decryption is kept secret and is not shared to everyone (Stallings, 2003). RSA is a Block cipher in which every message is mapped to an integer. Here, the public key can be known to everyone and is used for encrypting messages. The basic steps of this algorithm are: key generation, encryption and decryption (Stallings, 2003). The algorithm involves multiplying two large prime numbers and through additional operations deriving a set of two numbers that constitutes the public key and another set that is the private key (Khanna and Jaiswal, 2013). Using this RSA system the private key never needs to be sent across the internet. The private key is used to decrypt text that has been encrypted with the public key. Here, both the public and private keys are needed for encryption/decryption but only the owner of a private key ever needs to know it (Khanna and Jaiswal, 2013).

DATA ENCRYPTION TECHNIQUES

Enhancing of data security: This study explains the concept of accessing the cloud based application that eliminate the concerns regarding data privacy, segregation and provides different encryption algorithms to enhance security in cloud (Kaur and Mahajan, 2012). Here, user can select their choice of algorithm such as AES, DES, RSA and Blowfish base on their need and accordingly encrypt/decrypt data on cloud. In this, two public/private keys are used for encryption/decryption. The advantage of this method is user can dynamically select their own required algorithms but in this only technical privacy and encryption controls were analyzed (Kaur and Mahajan, 2012) (Table 2).

Table 2: Comparison of encryption techniques

Research	Encryption techniques	Features	Limitations
Kaur and Mahajan (2012)	AES, DES, RSA and Blowfish	User can dynamically select their own required algorithms	Only technical privacy and encryption controls were analyzed
Kalpana and Singaraju (2012)	RSA algorithm	Only authorized user can access the data, by this way it provides security	Key management complexity is high
Tebaa *et al.* (2012)	Homomorphic Encryption Method	Outsource the calculations on confidential data to cloud server and computation complexity reduced.	High cost
El-Etriby *et al.* (2012)	RC4, RC 6, MARS, AES, DES, 3 DES, Two fish and Blowfish	More suitable when we focus on time of encryption method	No strong indications of statistical weakness for eight modern Encryption algorithm in both environments
Lokhande and Kumari (2012)	Efficient encryption and decryption	Better security provided by separating encryption/decryption service from storage device	High cost due to multiple service provider
Meissen (2012)	Craig Gentry's homomorphic scheme and bootstrapping	It do not provide a potential attacker with any additional information. Unlimited usability of the cipher text.	Limitation on the arithmetic circuits It cannot properly decrypt the cipher text
Chung *et al.* (2014)	Key policy and cipher text policy	Reduces the computation overhead of the data owner. The sensitive information cannot be revealed	Hard to satisfy the property of multi-use and re-encrypt control. Cannot withstand the collusion attack
Song *et al.* (2013)	Chaos block Encryption algorithm and homomorphic Signature algorithm are used	Fast encryption/decryption speed. Strong confidentiality and strong data Low communication and computational costs	More complex and difficult than traditional computing environment
Gaurha and Shrivastava (2012)	Enhanced Complete Alu Sequence algorithm	It increases the cloud efficiency. Reduce the load distribution	It does not provide perfect encryption information. Less data encryption process and security
Kumar and Venkateswarlu (2013)	ABE (Attribute Based encryption) and FHE (Fully Homomorphic Encryption). LP (Linear Programming)	Secured LP computation. It fulfils input/output privacy, cheating resilience and efficiency computing resources is needed	No sensitive information can be derived during LP. Large amount of
Wang *et al.* (2009)	Considering different issues like byzantine failure, malicious data modification attacks and attacks from cloud server	It provides the localization of data error. Highly efficient and resilient	It does not focus on dynamic data operations. Drastically limited applicable in cloud data storage
Wang *et al.* (2011)	HABE (Hierarchical Attribute-Based Encryption) Model, a HABE scheme and a revocation mechanism	Collusion resistant and full delegation	Unauthorized access control of encrypted data. Revoking the access rights
Jachak *et al.* (2012)	Framework of a very light-weight and provably secure provable data possession scheme	It supports dynamic operations on data block. Very low cost	It violates the privacy-preserving guarantee. Large communication overhead and time delay

Table 2: Continue

Research	Encryption techniques	Features	Limitations
Jain and Kaur (2012)	Analyses the data security risk, the data security requirements, deployment of security functions	Acceptably secure. No danger of any data sent within the system	As computing power increases, the level of encryption is stepped up
Kamara and Raykova (2013)	Outsourcing computation to a cluster of machines which typically happens when the computation needs to be performed over massive datasets	It can reduce the client's work. It is quick for working with massive datasets	Large-scale clusters
Bouti and Keller (2012)	Homomorphic properties of Asymmetric Encryption algorithms Asymmetric Encryption algorithms addition and multiplication	It increases the security. Computation overhead can be reduced	In Existing method there may loss of confidentiality
Singh and Maini (2011)	Blowfish, AES and DES	DES and 3DES are known to have worm holes in their security mechanism	Blowfish and AES do not have any so far
Abdel-Karim	Blowfish, AES and DES	3DES has almost 1/3 throughput of DES or in other words it needs 3 times than DES to process the same amount of data	AES showed poor performance results compared to other algorithms
Singha and Raina (2011)	Blowfish , AES, DES and RC4	RC4 is fast in nature and consume less power with respect to its counterparts	Different modes of AES, throughput decreases as key size increases because of more usage of computational power
Singh *et al.* (2011)	Throughput analysis of blowfish, AES, DES and RC4 algorithms	Blowfish has better performance than other algorithms followed by AES in terms of throughput	3DES has least efficient of all the studied algorithms
Seth and Mishra (2011)	Comparative analysis of blowfish, AES, DES and RC4 algorithms	DES algorithm consumes least encryption time	RSA consume longest encryption time and memory usage is also very high
Elminaam *et al.* (2010)	Symmetric Encryption algorithms like AES (Rijndael), DES, 3DES, RC2 and blowfish	The RC6 requires less time	RC2 has low performance and low throughput
Idrus *et al.* (2008)	Performance analysis of blowfish, AES, DES and RC4 algorithms	Encryption delay overhead using AES is less than the overhead using RC4 and XOR algorithm	RC2 has low performance and low throughput
Gast (2002)	Performance analysis of blowfish, AES, DES and RC4 algorithms	encryption delay overhead using AES is less than the overhead using RC4 and XOR algorithm	RC2 has low performance and low throughput
Agrawal and Mishra (2012)	Symmetric key encryption algorithms DES, TRIPLE DES and AES	Symmetric key encryption is superior than asymmetric key encryption	RSA is slower
Elminaam *et al.* (2009)	Symmetric Encryption algorithms	Symmetric algorithms is lesser than Asymmetric Encryption algorithms	No significant difference when the results are displayed either in hexadecimal base encoding or in base 64 encoding
Soubhagya *et al.* (2013)	PHRs (Personal health record) which is stored in cloud servers are fully controlled by the patient	Prevent accidental leakage of private information. It supports efficient user revocation and access	Risks of privacy exposure and scalability in key management
Xiao *et al.* (2012)	Shamir's Secret Sharing algorithm and Rabin's Information Dispersal algorithm (IDA)	Computationally inexpensive. Able to reconstruct the whole data even when slices of data were unavailable	Attacker gets hold of the secrets easily. Space complexity is high
Gentry (2010)	Non-circuit based symmetric-key homomorphic encryption scheme	Realistic performance and simple algorithm Fully harmonic	Not semantically secure and time complexity is high
Chauhan and Vatta (2013)	Fully homo morphic encryption scheme	Consistent with privacy and highly parallelizable	Output may be significantly smaller bits
Inbarani *et al.* (2013)	Proxy re-encryption, type based proxy re-encryption, key private proxy re-encryption, identity based proxy re-encryption, attribute based proxy re-encryption and threshold proxy re-encryption	PRE is secure against plain text attack. Secure against an adaptive chosen cipher text attack	Average efficiency and flexibility. More difficult than that of CPA security. Collusion problem and plaintext attack
Zhang *et al.* (2012)	Practical message attack against all fully homomorphic encryption schemes	Useful in the outsourced computation scenario. Malicious cloud can recover the message	Very expensive and incapability of detecting the attack

RSA algorithm: This study ensuring the security of data in cloud by implementing RSA algorithm. Here, every message is mapped to an integer (Kalpana and Singaraju, 2012). Encryption is done by cloud service provider and

decryption is done by the cloud user or consumer. The main advantage of this is only authorized user can access the data, by this way it provides security. But here the key management complexity is high (Kalpana and Singataju, 2012).

Homomorphic encryption: This study proposes an application of a method to execute operations on encrypted data without decrypting them. Homomorphic Encryption Method is able to perform operations of encrypted data. The advantages of this method are: outsource the calculations on confidential data to cloud server and computation complexity reduced (Tebaa *et al*., 2012).

Modern encryption technique: This study presents an evaluation for selected eight modern encryption techniques namely RC4, RC 6, MARS, AES, DES, 3 DES, Two fish and blowfish in two different independent platforms namely desktop computer and Amazon EC 2 (El-Etriby *et al*., 2012). According to randomness testing using NIST statistical testing. There are three parameters p-value, rejection rate and time consuming taken for performance evaluation. AES Encryption Method is suitable algorithm for Amazon EC 2 but blowfish and DES is more suitable when we focus on time of encryption method. RC 6 Encryption Method is Suitable algorithm for traditional PC environment. But Blowfish is more suitable when we focus on time of Encryption Method (El-Etriby *et al*., 2012). But here no strong indications of statistical weakness for eight modern encryption algorithm in both environments.

Efficient encryption and decryption: This study proposes an efficient encryption and decryption service from the storage service of data. In here, service provider operates the encryption and decryption system, other provider operate the storage and application system (Lokhande and Kumari, 2012). The main purpose of this method is better security provided by separating encryption/decryption service from storage device. But high cost due to multiple service provider (Lokhande and Kumari, 2012).

A mathematical approach to fully homomorphic encryption: The homomorphic properties of various encryption schemes have been a fascination of the cryptographic community for decades. With the rise of cloud computing and decentralized processing, the need for security in such applications is increasing (Meissen, 2012). Only recently, however, has the construction of a fully homomorphic encryption scheme been realized. In this study, it present a mathematical approach to Craig Gentry's proposed fully homomorphic scheme. Then, start with an overview of other homomorphic encryption schemes, followed by an examination of polynomial rings and their relation to lattices. In the lattices a basis reduction algorithms is used and that algorithm is Lenstra-Lenstra-Lovasz (LLL) algorithm (Meissen, 2012). They introduce a method called bootstrapping for refreshing the cipher text allows us to perform an unlimited number of operations. It has two key concepts, namely operations and noise. Finally, explore the scheme itself and provide a foundation from which to understand the challenges faced when constructing a fully homomorphic encryption scheme. The advantages of this method are: it do not provide a potential attacker with any additional information and unlimited usability of the cipher text. But it has limitation on the arithmetic circuits and it cannot properly decrypt the cipher text (Meissen, 2012).

Attribute-based proxy re-encryption scheme: Attribute-Based Proxy Re-Encryption (ABPRE) scheme is one of the proxy cryptography which can delegate the re-encryption capability to the proxy and re-encrypt the encrypted data by using the re-encryption key. In this study, researchers survey two various access policy attribute-based proxy re-encryption schemes (key policy and cipher text policy) and analyse these schemes (Chung *et al*., 2014). The survey starts from an atomic proxy cryptography that can delegate the semi-trusted party to re-encrypt the encrypted data, followed by the first notion of the attribute-based proxy re-encryption based on cipher text policy. ABPRE contains five algorithms: KeyGen(), Encrypt(), ReKeyGen(), ReEncypt(), and Decrypt(). It has some criteria's namely; unidirectionality, data confidentiality, non-interactive, non-transitive, multi-use, re-encryption control, master key security and collusion resistant. These criteria's has to satisfy the following schemes; Blaze's scheme, Xiao *et al*. (2012)'s scheme, Luo's scheme, Yu's scheme, Yu's scheme, Do's scheme and Seo's scheme. Thereafter, an attribute-based proxy re-encryption with a constant number of paring operations is described at the end (Chung *et al*., 2014).

The advantages of this method are: it reduces the computation overhead of the data owner, user just uses his own secret key to decrypt the encrypted data and he doesn't need to store an additional decryption key for deciphering. The sensitive information cannot be revealed to the proxy in re-encryption and the proxy only complies

with the data owner's command (Chung *et al*., 2014). The major disadvantages are: hard to satisfy the property of multi-use and re-encrypt control and it cannot withstand the collusion attack by using the cloud and the data user (Chung *et al*., 2014).

An efficient encryption and verification scheme for preserving electronic evidence: This study proposes an efficient encryption and verification scheme to preserve electronic evidence in cloud computing. In this scheme, it constructs a Chaos Block Encryption Algorithm (CBEA) to encrypt electronic evidence and sends cipher evidence to the Cloud Storage Server (CSS) (Song *et al*., 2013). Then, it designs a Homomorphic Signature Algorithm (HSA) to establish a challenge and verification protocol and the user can use this protocol to verify that the CSS holds cipher evidence intact. Experiment and security analysis show this scheme can ensure the confidentiality of electronic evidence in cloud with fast encryption/decryption speed and can offer a double integrity verification guarantee for electronic evidence in cloud with low communication and computational costs (Song *et al*., 2013). The main advantages of this technique are: fast encryption/decryption speed, strong confidentiality and strong data possession and low communication and computational costs. But this technique is more complex and difficult than traditional computing environment (Song *et al*., 2013).

Data security in cloud computing using linear programming: This study presents system which implements robust design with perfect security constraints. This method implements verification mechanism for all kind of customers and all kinds of feasible solution in different number of ways under Linear Programming (LP) conditions (Gaurha and Shrivastava, 2012). This mechanism brings cloud customer great computation savings from secure LP outsourcing as it only incurs for some local computation overhead on the customer while solving a normal LP problem usually requires more than time. This technique can always help customers achieve >30×savings when the sizes of the original LP problems are not too small while introducing no substantial overhead on the cloud. LP contains perfect design framework. It can contains some of the steps are available in design framework Key generation, probability encryption, proof generation and result decryption. By using fully Homomorphic Encryption Method, data become more secure. Here, enhanced complete Alu Sequence algorithm is used. This mechanism design is able to explore appropriate security/efficiency trade-off via higher level LP computation than the general circuit representation (Gaurha and Shrivastava, 2012).

The main advantages of this programming are: it fulfils input/output privacy, cheating resilience and efficiency: it increases the cloud efficiency and it reduces the load distribution. The disadvantages are: it does not provide perfect encryption information: it do not work properly under the verification representation process and less data encryption process and security (Gaurha and Shrivastava, 2012).

Efficiently providing data security and linear programming in cloud computing: This study presents a new system for searching on encrypted data which combined ABE (Attribute Based Encryption) and FHE (Fully Homomorphic Encryption) (Kumar and Venkateswarlu, 2013). This system enables anyone even without private-key of the encrypted data to search the data and it formalize the problem of securely outsourcing LP (Linear Programming) computations in cloud computing and provide such a practical mechanism design which fulfils input/output privacy, cheating resilience and efficiency. Here, the following algorithms are used KeyGen(1k)→{K}. ProbEnc(K,_)→{_K}. ProofGen(_K)>{(y, Φ)}. Result Dec (K, _, y, Φ) > {x,⊥}. The merits of this technique are secured LP computation and it fulfils input/output privacy, cheating resilience and efficiency. But no sensitive information can be derived during LP and it needs large amount of computing resources (Kumar and Venkateswarlu, 2013).

Ensuring data storage security in cloud computing: This study explains about a concept of implementing a new method to provide security for data in cloud computing by considering different issues like byzantine failure, malicious data modification attacks and attacks from cloud server. Here, mainly concentrate on security issues and explain about using homomorphic token with distributed verification of erasure code data (Wang *et al*., 2009). Using homomorphic token improves security in terms of finding out misbehaving servers, security operations on data blocks including security for updating, deleting and modifying data. The merits of this method are: it provides the localization of data error and highly efficient and resilient. But it does not focus on dynamic data operations and it is drastically limited applicable in cloud data storage (Wang *et al*., 2009).

Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers: This study proposes a HABE (Hierarchical Attribute-Based Encryption) Model, a HABE scheme and a revocation mechanism, so as to simultaneously achieve: high

performance; fine-grained access control; scalability and full delegation, in cloud computing (Wang *et al*., 2011). It describes the scalable revocation scheme by applying Proxy Re-Encryption (PRE) and Lazy Re-Encryption (LRE) to the HABE scheme from the systematic point of view and prove the HABE scheme which is also collusion resistant, to be semantically secure against adaptive chosen plaintext attacks under the BDH assumption and the Random Oracle Model (Wang *et al*., 2011). The advantages of this method are: collusion resistant and full delegation. But demerits are: unauthorized access control of encrypted data and revoking the access rights (Wang *et al*., 2011).

Homomorphic authentication with random masking technique ensuring privacy and security in cloud computing: This study proposes a framework of a very light-weight and provably secure provable data possession scheme. It studies the problem of ensuring the integrity and security of data storage in cloud computing. Security in cloud is achieved by signing the data block before sending to the cloud (Jachak *et al*., 2012). Signing is performed using Boneh-Lynn-Shacham (BLS) algorithm which is more secure compared to other algorithms. It surpasses prior work on several counts including storage, bandwidth and computation overheads as well as the support for dynamic operations. This method uses a challenge-response protocol to ensure the intactness of data and framework fully supports dynamic operations on data block which are very efficient (Jachak *et al*., 2012). So, the intactness of data is verified and along with that it provides protection against server colluding attacks which are more difficult to deal with. The advantages are: it supports dynamic operations on data block, it protects against server colluding attacks and very low cost. The disadvantages are: it violates the privacy-preserving guarantee and large communication overhead and time delay (Jachak *et al*., 2012).

Implementing DES algorithm in cloud for data security: This study proposes that the cloud data security must be considered to analyse the data security risk, the data security requirements, deployment of security functions and the data security process through encryption. The main contribution of this study is the new view of data security solution with encryption which is important and can be used as reference for designing the complete security solution (Jain and Kaur, 2012). The security architecture of the system is designed by using DES cipher block chaining which eliminates the fraud that occurs today with stolen data. In order to be secure the system the communication between modules is encrypted using symmetric key. The advantages are: acceptably secure and no danger of any data sent within the system. The disadvantage is as computing power increases; the level of encryption is stepped up (Jain and Kaur, 2012).

Parallel homomorphic encryption: This research considers the problem of privately outsourcing computation to a cluster of machines which typically happens when the computation needs to be performed over massive datasets. At such scales, computation is beyond the capabilities of any single machine so it is performed by large-scale clusters of workers (Kamara and Raykova, 2013). To address this problem, it considers the Parallel Homomorphic Encryption (PHE) schemes which are encryption schemes that support computation over encrypted data through the use of an Evaluation algorithm that can be efficiently executed in parallel. Then focus on the Map Reduce Model of parallel computation and show how to construct PHE schemes that can support various map reduce operations on encrypted datasets including element testing and keyword search. Underlying this PHE schemes are two new constructions of (local) Randomized Reductions (RR) for univariate and multivariate polynomials (not based on secret sharing and are fully-hiding in the sense that the privacy of the input is guaranteed even if the adversary sees all the client's queries) (Kamara and Raykova, 2013). The randomized reduction for univariate polynomials is information-theoretically secure and is based on permutation polynomials whereas this reduction for multivariate polynomials is computationally-secure under the multi-dimensional noisy curve reconstruction assumption. The main uses of this method are: it can reduce the client's work, it is secure against single-input chosen-plaintext and function attacks and it is quick for working with massive datasets. The disadvantages are: large-scale clusters and the client has to run the (randomized reductions) RR's Recovery algorithm which can represent a non-trivial amount of research (Kamara and Raykova, 2013).

Securing cloud-based computations against malicious providers: This study presents an approach to exploit homomorphic properties of Asymmetric Encryption algorithms to increase the security and the trust level in today's cloud based services. This approach is based on the use of two different Asymmetric Encryption algorithms addition and multiplication (Bouti and Keller, 2012). Here, a protocol to delegate computations into clouds with encrypted data is used. The protocol is implemented in user mode and does not require any changes to the operation system. The protocol is based

on homomorphic properties of Encryption algorithms. The protocol can also be used to amend existing applications by software patches of binaries. This study evaluates the protocol by a proof-of-concept implementation to investigate practicability and discuss variants and extensions to increase the prototype's efficiency (Bouti and Keller, 2012). The security of the proposed protocol depends on the security of the employed cryptosystems, in the case El-Gamal and Paillier that are semantically secure under chosen plaintext attacks and thus ensure the confidentiality of the data during transmission and computation against passive attacks. Another side effect of the algorithms used that may influence the security of the presented protocol is the so called malleability property of paillier cryptosystem. The presented protocol can be implemented in application source code. Existing applications with no possibility to modify the source code might be extended by the functionality through software patches of the binary executable, although the possibility to reduce the number of transfers is strongly reduced (Bouti and Keller, 2012). The advantages are: it increases the security, secure delegated computation and computation overhead can be reduced. But in existing method there may loss of confidentiality.

Comparison of data Encryption algorithms: This method proposes comparison of Data Encryption algorithms. Blowfish has better performance than other commonly used encryption algorithms. AES showed poor performance results compared to other algorithms since it requires more processing power. In this study, the first set of experiments were conducted using ECB mode. The results show the superiority of Blowfish algorithm over other algorithms in terms of processing time. AES consumes more resources when data block size is relatively big. Here, that 3DES requires always more time than DES because of its triple phase encryption characteristic. Blowfish which has a long key (448 bit), outperformed other Encryption algorithms. DES and 3DES are known to have worm holes in their security mechanism; Blowfish and AES do not have any so far (Singh and Maini, 2011).

Performance analysis of Data Encryption algorithms: This study shows that blowfish has a better performance than other Common Encryption algorithms used. Since, Blowfish has not any known security weak points which makes it an excellent candidate to be considered as a Standard Encryption algorithm. AES showed poor performance results compared to other algorithms since, it requires more processing power. Using CBC mode has added extra processing time but overall it was relatively negligible especially for certain application that requires more secure encryption to a relatively large data blocks. The blowfish has a very good performance compared to other algorithms. Also, it showed that AES has a better performance than 3DES and DES. Amazingly it shows also that 3DES has almost 1/3 throughput of DES or in other words it needs 3 times than DES to process the same amount of data (Verma *et al*., 2011).

Comparative analysis of AES and RC4 algorithms for better utilization: This study shows that comparative analysis of AES and RC4 algorithms for better utilization has designed. The performance metrics were throughput, CPU process time, memory utilization, encryption and decryption time and key size variation. The RC4 is fast and energy efficient for encryption and decryption and it is better than AES. When compare the encryption time of AES and RC4 algorithm over different packet size, RC4 takes less time to encrypt files with respect to AES. In AES, CFB and CBC takes nearly similar time but ECB takes less time then both of these (Singha and Raina, 2011).

Another performance comparison point is the changing key size. The three different key sizes used are 128, 192 and 256 bits. As the key size vary from 128-192 bits to 256 bits, encryption time for RC4 is almost constant and is less then AES. Hence, it consumes less power with respect to AES. But for different modes of AES, encryption time increases as key size increases. The result shows the superiority of RC4 over AES. With different key sizes RC4 gives almost the same result. But for different modes of AES, throughput decreases as key size increases because of more usage of computational power and encryption characteristics. Thus, RC4 is fast in nature and consume less power with respect to its counterparts (Singha and Raina, 2011).

Through put analysis of various Encryption algorithms: In this study, the throughput analysis of various Encryption algorithms presented. The throughput of encryption as well as decryption schemes is calculated but one by one. In the case of encryption scheme throughput is calculated as the average of total plain text in k bytes divided by the average encryption time and in the case of decryption scheme throughput is calculated as the average of total cipher text is divided by the average decryption time. This research presents the performance evaluation of selected Symmetric algorithms. The Selected algorithms are AES, 3DES, Blowfish and DES. Finally concluded that Blowfish has better performance than

other algorithms followed by AES in terms of throughput. Secondly 3DES has least efficient of all the studied algorithms (Singh *et al*., 2011).

Comparative analysis of Encryption algorithms for data communication: This study proposes comparative analysis of Encryption algorithms for data communication. This study analyse the performance of Encryption algorithm is evaluated considering the following parameters like computation time, memory usage and output bytes. The comparison of three algorithm AES, DES and RSA using same text file for five experiment, output byte for AES and DES is same for different sizes of files. The RSA has very smaller output byte compare to AES and DES algorithm. Time taken by RSA algorithm is much higher compare to the time taken by AES and DES algorithm. Based on the text files used and the experimental result it was concluded that DES algorithm consumes least encryption time and AES algorithm has least memory usage while encryption time difference is very minor in case of AES algorithm and DES algorithm. RSA consume longest encryption time and memory usage is also very high but output byte is least in case of RSA algorithm (Seth and Mishra, 2011).

Evaluating the performance of symmetric Encryption algorithms: This study presents the performance of Symmetric Encryption algorithms like AES (Rijndael), DES, 3DES, RC2, Blowfish and RC6. A comparison has been conducted for those Encryption algorithms at different settings for each algorithm such as different sizes of data blocks, different data types, battery power consumption, different key size and finally encryption/decryption speed (Elminaam *et al*., 2010). The RC6 requires less time than all algorithms except Blowfish. The AES has an advantage over other 3DES, DES and RC2 in terms of time consumption and throughput. The 3DES has low performance in terms of power consumption and throughput when compared with DES. It always requires more time than DES because of its triple phase encryption characteristics. Finally, it is found that RC2 has low performance and low throughput when compared with other five algorithms in spite of the small key size used (Elminaam *et al*., 2010).

Performance analysis of Encryption algorithms text length size on web browsers: This study proposes the work in the different browsers for evaluate the performance analysis of Encryption algorithms text length size. This study presents the study of security measure level for a web programming language to analyze four Web browsers (Idrus *et al*., 2008). This study consider of measuring the performances of encryption process at the programming language's script with the Web browsers. A study is conducted for different popular Secret Key algorithms such as RC4, AES and XOR. In this they were implemented and their performance was compared by encrypting for real time video streaming of varying contents. The results showed: encryption delay overhead using AES is less than the overhead using RC4 and XOR algorithm. Therefore, AES is a feasible solution to secure real time video transmissions (Gast, 2002).

Comparative survey on symmetric key encryption techniques: This study proposes the popular Symmetric Key Encryption algorithms such as DES, TRIPLE DES, AES and Blowfish. Symmetric Key algorithms run faster than Asymmetric Key algorithms such as RSA, etc. and the memory requirement of Symmetric algorithms is lesser than Asymmetric Encryption algorithms. Further, the security aspect of symmetric key encryption is superior than asymmetric key encryption (Agrawal and Mishra, 2012).

Performance evaluation of Symmetric Encryption algorithms: This study analyses the Performance Evaluation of Symmetric Encryption algorithms. Researchers use a laptop IV 2.4 GHz CPU, in which performance data is collected. In the experiments, the laptop encrypts a different file size ranges from 321 kB to 7.139 MB. Several performance metrics are collected: encryption time, CPU process time and CPU clock cycles and battery power (Elminaam *et al*., 2009).

The selected algorithms are AES, DES, 3DES , RC6, Blowfish and RC2. Several points can be concluded from the simulation results. First; there is no significant difference when the results are displayed either in hexadecimal base encoding or in base 64 encoding (Elminaam *et al*., 2009).

A homomorphic encryption technique for scalable and secure sharing of personal health record: In this study, a mechanism for secure data sharing, access control to PHRs (Personal Health Record) which is stored in cloud servers are fully controlled by the patient. A high degree of patient privacy is ensured by exploiting homomorphic encryption technique. For secure data outsourcing, the users are divided in the PHR System into multiple security domains that greatly reduces the key management for owners and users. It proposes mechanisms for key distribution and encryption so that PHR owners can specify personalized fine-grained role-based access

policies during file encryption. It provides a thorough analysis of the complexity and scalability of the proposed secure PHR sharing solution (Soubhagya *et al*., 2013).

The advantages of this method are prevent accidental leakage of private information, high degree of patient privacy is ensured and it supports efficient user revocation and access control. The main drawbacks are risks of privacy exposure, scalability in key management and flexible access and efficient user revocation (Soubhagya *et al*., 2013).

A comparative study of the Secret Sharing algorithms for secure data in the cloud: In this study, the performance of two secret sharing algorithms are compared. The Shamir's Secret Sharing algorithm and Rabin's Information Dispersal Algorithm (IDA) are implemented in a private cloud setup using the open stack cloud framework (Nirmala *et al*., 2012). Data encryption, homomorphic encryption, Secret Sharing algorithms and Private Information Retrieval (PIR) are the techniques widely used for secure data outsourcing. CIA (Confidentiality, Integrity and Availability) are the challenging issues associated with data storage management with/without data outsourcing. Using the Shamir's Secret Sharing and the Rabin's IDA, transform and distribute the employee sample database to three virtual machines within 47 and 30 sec, respectively (Nirmala *et al*., 2012). The advantages are: computationally inexpensive and able to reconstruct the whole data even when slices of data were unavailable. The disadvantages are: attacker gets hold of the secrets easily and space complexity is high.

An efficient homomorphic encryption protocol for multi-user systems: This study develops a non-circuit based symmetric-key homomorphic encryption scheme. It is proven that the security of our encryption scheme is equivalent to the large integer factorization problem and it can withstand an attack with up to in poly chosen plaintexts for any predetermined where the security parameter is. It is further considers practical multiple-user data-centric applications (Xiao *et al*., 2012). Here, this method proposes to transform the master encryption key into different user keys and develop a protocol to support correct and secure communication between the users and the server using different user keys. In order to prevent collusion between some user and the server to derive the master key, one or more key agents can be added to mediate the interaction. The advantages are, realistic performance and simple algorithm, fully homomorphic and possibility of encrypting data without needing to know the private key. But it is not semantically secure and time complexity is too high for practical use (Xiao *et al*., 2012).

Computing arbitrary functions of encrypted data: This study describes a "fully homomorphic" encryption scheme that keeps data private but that allows a worker that does not have the secret decryption key to compute any (still encrypted) result of the data, even when the function of the data is very complex. A third party can perform complicated processing of data without being able to see it. The merits are, consistent with privacy and highly parallelizable. But output may have significantly fewer bits (Gentry, 2010).

Cyber security in data mining using homomorphic encryption: This study discusses the issue of privacy preserving data mining and presents the technique that provides the privacy on data mining application (Chauhan and Vatta, 2013). It uses the asymmetric encryption to provide the privacy and used the RSA encryption to encrypt the data and it also presents a client server architecture that connects to the multiple clients. This proposed protocol is to encrypt the data so it uses the encryption technique to encrypt the data and homomorphic encryption to secure the information (Chauhan and Vatta, 2013).

The main advantages are: without security the data may stand compromised and it is used in real time applications. But it refreshes the data only on the server site and it has the fixed length.

Proxy re-encryption schemes for data storage security in cloud a survey: This study surveys different proxy re-encryption schemes used in Cloud Storage System. To keep the sensitive user data confidential against untrusted servers several proxy re-encryption techniques are used. It explores various data encryption techniques such as proxy re-encryption, type based proxy re-encryption, key private proxy re-encryption, identity based proxy re-encryption and attribute based proxy re-encryption and threshold proxy re-encryption (Inbarani *et al*., 2013).

The advantages are: PRE is secure against plain text attack, semantics security and cipher text privacy control. It provides CCA security, secure against an adaptive chosen cipher text attack, fine-grained access control on encrypted data, security against chosen cipher text attack, scalable user revocation and data forwarding. The disadvantages are: collusion problem and plaintext attack, encoding operations over encrypted messages is not

possible. The key privacy proof is more difficult than that of CPA security, difficult to find efficient constructions for multiuse CCA-secure IBE-PRE, average efficiency and flexibility. It is difficult to design CCA secure C-PRE scheme, requires effective time period to be the same for all attributes associated with the user and high access control (Inbarani *et al*., 2013).

Reaction attack on outsourced computing with fully homomorphic encryption schemes: This study proposes a reaction attack against full homomorphic schemes when they are used for securing outsourced computation. Essentially, this attack is based on the user's reaction towards the output generated by the cloud and this attack enables us to retrieve the associated secret key of the system (Zhang *et al*., 2012). The homomorphic encryption schemes, although seem to be a promising candidate have some problems when they are used in the context of cloud computing. Here, it uses a practical message attack against all fully homomorphic encryption schemes, in that a malicious cloud can recover the messages by observing users reactions (Zhang *et al*., 2012).

The main advantages of this method are: malicious cloud can recover the message by using fully homomorphic scheme and it is useful in the outsourced computation scenario. The disadvantages are: it has some problems while using in the context of cloud computing, it is very expensive and user is incapable of detecting the attack.

CONCLUSION

In this study, rigorous analysis is made on data encryption techniques which are used to convert the plain text data into cipher text data in the cloud environment. Many encryption techniques have been analysed such as RC4, RC6, MARS, AES, DES, 3DES, RSA two fish and Blowfish. The ultimate goal of these techniques is to provide high security and efficiently support for large scale and distributed nature of cloud data. In today's world, demand of cloud is increasing so the security of cloud and consumer is very important. These techniques are helpful for accessing the applications in cloud environment.

REFERENCES

Agrawal, M. and P. Mishra, 2012. A comparative survey on symmetric key encryption techniques. Int. J. Comput. Sci. Eng., 4: 877-882.

Bouti, A. and A.J. Keller, 2012. Securing cloud-based computations against malicious providers. J. ACM SIGOPS Operating Syst., 46: 38-42.

Buyya, R., C. Vecchiola and S.T. Selvi, 2013. Mastering Cloud Computing: Foundations and Applications Programming. 1st Edn., Morgan Kaufmann Publishers, ISBN-13: 978-0124114548, UK., Pages: 468.

Chauhan, E. and S. Vatta, 2013. Cyber security in data mining using homomorphic encryption. Int. J. Adv. Res. Comput. Sci. Software Eng., 3: 1147-1150.

Chung, P.S., C.W. Liu and M.S. Hwang, 2014. A study of attribute-based proxy re-encryption scheme in cloud environments. Int. J. Network Secur., 16: 1-13.

El-Etriby, S., E.M. Mohamed and H.S. Abdul-Kader, 2012. Modern encryption techniques for cloud computing: Randomness and performance testing. Proceedings of the 2nd International Conference on Communications and Information Technology, February 2012, Cambridge, UK., pp: 800-805.

Elminaam, D.S.A., H.M. Abdual-Kader and M.M. Hadhoud, 2010. Evaluating the performance of symmetric encryption algorithms. Int. J. Network Sec., 10: 216-222.

Elminaam, D.S.A., H.M. Abdul Kader and M.M. Hadhoud, 2009. Performance evaluation of symmetric encryption algorithms. Commun. IBIMA, 8: 58-64.

Gast, M.S., 2002. 802.11 Wireless Networks: The Definitive Guide. O'Reilly Publisher, UK., pp: 70-76.

Gaurha, N. and M. Shrivastava, 2012. Data security in cloud computing using linear programming. Int. J. Emerging Technol. Adv. Eng., 2: 28-30.

Gentry, C., 2010. Computing arbitrary functions of encrypted data. J. Commun. ACM, 53: 97-105.

Idrus, S.Z.S., S.A. Aljunid, S.M. Asi, S. Sudin and R.B. Ahmad, 2008. Performance analysis of encryption algorithms' text length size on web browsers. Int. J. Comput. Sci. Network Secur., 8: 20-25.

Inbarani, W.S., G. Shenbagamoorthy and C.K.C. Paul, 2013. Proxy re-encryption schemes for data storage security in cloud- a survey. Int. J. Eng. Res. Technol., 2: 1-5.

Jachak, K.B., S.K. Korde, P.P. Ghorpade and G.J. Gagare, 2012. Homomorphic authentication with random masking technique ensuring privacy and security in cloud computing. J. Bioinfo Secur. Inform., 2: 49-52.

Jain, N. and G. Kaur, 2012. Implementing DES algorithm in cloud for data security. VSRD Int. J. Comput. Sci. Inform. Technol., 2: 316-321.

Kalpana, P. and S. Singaraju, 2012. Data security in cloud computing using RSA algorithm. Int. J. Res. Comput. Commun. Technol., 1: 143-146.

Kamara, S. and M. Raykova, 2013. Parallel Homomorphic Encryption. In: Financial Cryptography and Data Security, Adams, A.A., M. Brenner and M. Smith (Eds.). Springer-Verlag, Berlin, Heidelberg, ISBN: 978-3-642-41319-3, pp: 213-225.

Kaur, M. and M. Mahajan, 2012. Implementing various encryption algorithms to enhance the data security of cloud in cloud computing. Int. J. Comput. Sci. Inform. Technol., 2: 831-835.

Khanna, L. and A. Jaiswal, 2013. Cloud computing: Security issues and description of encryption based algorithms to overcome them. Int. J. Adv. Res. Comput. Sci. Software Eng., 3: 279-283.

Kumar, S.K. and S. Venkateswarlu, 2013. Efficiently providing data security and linear programming in cloud computing. Int. J. Comput. Sci. Res. Technol., 4: 630-632.

Lokhande, V. and P.P. Kumari, 2012. Efficient encryption and decryption services for cloud computing. Int. J. Social Appl. Comput. Sci., 1: 71-75.

Meissen, R., 2012. A mathematical approach to fully homomorphic encryption. Ph.D. Thesis, Worcester Polytechnic Institute.

Nirmala, S.J., S.M.S. Bhanu and A. Akhtar Patel, 2012. A comparative study of the secret sharing algorithms for secure data in the cloud. Int. J. Cloud Comput.: Serv. Archit., 2: 63-71.

Seth, S.M. and R. Mishra, 2011. Comparative analysis of encryption algorithms for data communication. Int. J. Comput. Sci. Telecommun., 2: 292-294.

Singh, G., A.K. Singla and K.S. Sandha, 2011. Through put analysis of various encryption algorithms. Int. J. Comput. Sci. Telecommun., Vol. 2.

Singh, S.P. and R. Maini, 2011. Comparison of data encryption algorithms. Int. J. Comput. Sci. Commun., 2: 125-127.

Singha, N. and J.P.S.Raina, 2011. Comparative analysis of AES and RC4 algorithms for better utilization. Int. J. Comput. Trends Technol., 2011: 177-181.

Song, X., H. Deng, L. Chen and M. Xiao, 2013. An efficient encryption and verification scheme for preserving electronic evidence in cloud computing. J. Inform. Comput. Sci., 10: 911-922.

Soubhagya, B., G.V. Mini and A.J. Jeya Celin, 2013. A homomorphic encryption technique for scalable and secure sharing of personal health record in cloud computing. Int. J. Comput. Appl., 67: 40-44.

Stallings, W., 2003. Cryptography and Network Security: Principles and Practice. 3rd Edn., Prentice Hall, London, UK., ISBN: 9780130914293, Pages: 681.

Tebaa, M., S. El Hajji and A. El Ghazi, 2012. Homomorphic encryption applied to the cloud computing security. Proceedings of the World Congress on Engineering, Volume 1, July 4-6, 2012, London, UK., pp: 1-4.

Verma, O.P., R. Agarwal, D. Dafouti and S. Tyagi, 2011. Peformance analysis of data encryption algorithms. Proceedings of the 3rd International Conference on Electronics Computer Technology, April 8-10, 2011, Kanyakumari, pp: 399-403.

Wang, C., Q. Wang, K. Ren and W. Lou, 2009. Ensuring data storage security in cloud computing. Proceedings of the 17th International Workshop on Quality of Service, July 13-15, 2009, Charleston, SC., USA., pp: 1-9.

Wang, G., Q. Liu, J. Wu and M. Guo, 2011. Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers. Comput. Secur., 30: 320-331.

Xiao, L., O. Bastani and I.L. Yen, 2012. An efficient homomorphic encryption protocol for multi-user systems. IACR Cryptology ePrint Archive, Pages: 193. https://eprint.iacr.org/2012/193.pdf.

Zhang, Z., T. Plantard and W. Susilo, 2012. Reaction attack on outsourced computing with fully homomorphic encryption schemes. Proceedings of the 14th International Conference on Information Security and Cryptology, November 30-December 2, 2011, Seoul, Korea, pp: 419-436.