CHAPTER 1

ENCRYPTION ALGORITHM FOR DATA SECURITY IN CLOUD COMPUTING

Anindita Desarkar[1], Ajanta Das[2]

[1] Department of Computer Science and Engineering, Jadavpur University, Kolkata, India
[2] Department of Computer Science and Engineering, University of Engineering & Management, Kolkata, Kolkata, India
Email: aninditadesarkar@gmail.com, cse.dr.ajantadas@gmail.com

Abstract

Cloud computing is the concept of using a virtual pool of resources to provide users with solutions to various computing problems via the internet. IT services are provided on an on-demand basis, which are accessible from anywhere, anytime through authorized users. "Storage as a Service" is one of the major services for the end users where sensitive data is stored in the cloud. As a result, data vulnerability becomes a common phenomenon where exploitation occurs through the provider or unauthorized users. So, data protection is the heart of data security where encryption algorithms play a major role. The greater complexity of these algorithms makes it more secure and safe compared to the other techniques. This chapter presents a few of the well-known encryption-decryption-based algorithms which are aimed at protecting cloud stored data from unauthorized access.

Keywords: Cloud computing, encryption algorithm, data security

1.1 Introduction

Cloud computing, which is the next-generation paradigm in computation, delivers applications and resources on an on-demand basis via the internet as services [1]. It provides an environment of hardware and software resources over the network to satisfy user requirements.

According to the National Institute of Standards and Technology (NIST) [2], cloud computing allows ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. Resources include computing applications, network resources, platforms, software services, virtual servers and computing infrastructure [3].

Computing and data storage are two basic functionalities provided by cloud computing. Cloud service consumers get the necessary access to their data and complete the computing job through the internet. They have no idea about the storage location of data and machine details which perform the computing task.

Data protection and security is the primary objective in data storage to gain the user's trust and make the implementation successful. Hence, data security is a burning issue in this domain, as data is scattered throughout various machines located in various locations. This makes it more complicated compared to the traditional systems. Though data security is the major issue, it's accompanied by several others like compliance, privacy, trust, and legal matters. Hence, adequate tools and techniques are required to be in place to make the cloud adoption initiative successful. In this chapter, various well-known techniques in cloud data security are reviewed for the purpose of achieving this goal.

Section 1.5.3 highlights existing research works in related areas. A brief overview of cloud computing is depicted in Section 1.5.4. Section 1.5.5 discusses various well-known techniques implemented in this domain. It discusses a few well-known algorithms from both domains – symmetric encryption and asymmetric encryption. Section 1.5.6 presents the comparison of these algorithms based on various parameters. Performance analysis of encryption algorithms in cloud is presented in Section 1.5.7. Section 1.5.8 contains the conclusions drawn on the basis of the above.

1.2 Related Work

Kartit *et al.* have reviewed the commonly used encryption algorithms for data security and proposed a simple, secure and privacy-preserving architecture for inter-cloud data sharing. This architecture is built on the concept of encryption and decryption algorithms, aimed at securing cloud data from unauthorized access [4]. A brief overview of various symmetric and asymmetric algorithms along with their comparison is presented by Bhardwaj *et al.* in their paper [5]. The research was enhanced by Iyer *et al.* [6], who have presented an algorithm which works towards providing a secure way to communicate and store data in cloud servers.

Conner *et al.* [7] have proposed an effective reputation management system with associated trust establishment by using multiple scoring functions and implemented the security service on a realistic application scenario in distributed environments. Friedman and West [8] and Ristenpart *et al.* [9] have presented several privacies as well as security issues that arise in a cloud computing framework. Yan *et al.* [10] described a nice scheme for handling data protection in terms of confidentiality by implementing amalgamation of identity

management with hierarchical identity-based cryptography for distribution of the key as well as mutual authentication in the cloud infrastructure. The security and privacy of data stored in cloud is the challenging task. Encryption algorithms are used for data security. In each algorithm an encryption key is used that can only be accessed by the authorized user. Ukil *et al.* [11] proposed an architecture and security model towards better protection of confidentiality and privacy in a public cloud infrastructure which does not depend on the deployment of the cloud.

1.3 Cloud Computing - A Brief Overview

Cloud computing refers to the delivery of all the computing services which majorly includes servers, storage, databases, networking and software over the internet for providing resource flexibility and lowering operating cost of the users. Lower cost, speed, global scale, productivity, performance and security are the top benefits of adopting this new technique over the traditional one. It eliminates or reduces the capital expense of buying necessary hardware and software, which works towards overall cost reduction. As most of the services are provided on demand, a huge amount of computing services can be arranged within a few minutes. It is also location independent because everything is accessible online. Optimized performance is achieved as the data centers, responsible for providing secure services, are updated with the latest generation of fast and efficient computing hardware. The following subsections describe its essential characteristics, various layers and commonly available deployment models [12].

1.3.1 Essential Characteristics

Cloud computing includes various unique characteristics, of which the following five are the primary ones.

- On-Demand Self-Service: An end user can get the required services automatically without human interaction with each service provider.
- Broad Network Access: Services are available over the network and accessed through standard mechanisms which encourage the use of heterogeneous thin or thick client platforms.
- Resource Pooling: The provider's computing resources are selected across the multiple consumers through a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. The customer has no or little knowledge about the location of the resources. However, they may be able to get generic information about it like country, state or datacenter. Examples of resources include storage, processing, memory and network bandwidth.
- Elasticity: Resource allocation can be increased or decreased based on the user's need or demand. The consumer gets the feeling of unlimited availability of resources as it can be arranged in any valid quantity at any time.
- Measured Service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth and active user accounts). Usage of resources is monitored, controlled and reported to provide transparency to the provider as well as the consumer.

1.3.2 Layers of Cloud Computing

Figure 1.1 below presents the three well-known layers of cloud computing: IaaS, PaaS and SaaS.

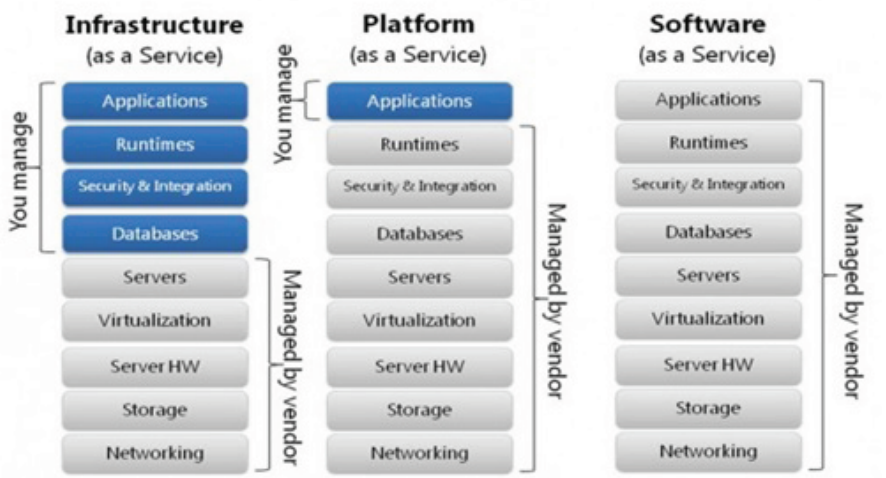

Figure 1.1 Layers of cloud computing.

Infrastructure as a Service (IaaS): IaaS is the option which provides only the base infrastructure. The end user needs to configure and manage the platform and environment and build all the required applications on top of it. Examples include AWS (EC2), GCP (CE) and Microsoft Azure (VM).

Platform as a Service (PaaS): PaaS is a cloud computing model where hardware and software tools are delivered by a third party provider. The user can build and manage applications without incurring the hazards of building and maintaining the infrastructure. Google App Engine, CloudFoundry, Heroku, and AWS (Beanstalk) are some examples of PaaS. The primary benefit of PaaS is its simplicity and convenience for users because they can access the infrastructure anywhere through a web browser. As a result, capital expenditure is removed or reduced to a great extent, which is traditionally required in the business. However, customers can face difficulties if providers experience the service outage or infrastructure disruption.

Software as a Service (SaaS): SaaS model allows providing software application as a service to the users. It basically refers to the concept of "software availability based on demand." Users generally access it through a thin client via a web browser. A few of the important characteristics include availability of software over the net, software maintenance by the vendors, subscription-based software licensing, centralized feature updating which mandates the user to download patches and upgrades. In this model, most of the services like applications, data, middleware, servers, storage and networking are maintained and managed by vendors and users only use it. Gmail is the most common example of this model.

1.3.3 Cloud Deployment Models

Following are the three majorly used deployment models which are commonly used across the world. The appropriate model should be chosen based on the organizational need.

- Private Cloud: A private cloud consists of resources which are exclusively available to a specific organization. The data center can be located on the company's onsite location or managed by third party service providers. Here all the services and infrastructure are maintained in a private network.
- Public Cloud: A public cloud is owned and maintained by third party cloud service providers where all the resources are available to multiple stakeholders or organizations. The service provider gives necessary access to the services based on the account created by the users. The user pays the cost based on the services they use.
- Hybrid Cloud: Hybrid cloud merges both the types – public and private – by the techniques which allow data and applications to be shared between them. The business receives greater flexibility, more deployment options, improvement in existing infrastructure, security and compliance by allowing the movement of data and applications between private and public clouds.

1.4 Data Security in Cloud Storage

Cloud storage is the convenient way to provide data access anytime and anywhere across the globe. Benefits brought by cloud storage majorly include scalability, accessibility and decreased IT overhead, which are also the driving factors of rapid adoption of this technology. So, there is a crying need for improved tools and techniques which keep sensitive data safe and secure in the cloud. Businesses and several enterprises have already adopted cloud services as it's able to provide cost-effective and flexible solutions as an alternative to expensive, locally implemented hardware. However, it also invites several unwanted risks as confidential files and sensitive data may be exposed to the outer world. So, enterprises always try to deploy new and improved technologies to secure cloud storage though initial measures which are implemented by cloud providers as a part of their solution. Authentication, access control and basic encryption techniques are included in the preliminary protection scheme. Subsection 1.5.5.1 of this chapter presents a few burning issues in the domain of cloud and Subsection 1.5.5.2 discusses a few such techniques which are used to increase cloud data security [13].

1.4.1 Security Issues in Cloud

Data security is a common concern for any technology, but it becomes a major challenge in the case of cloud infrastructure. Security concerns relate to risk areas such as external data storage, dependency on the "public" internet, lack of control, multi-tenancy and integration with internal security. Encryption techniques have been used for a long time to secure sensitive data. Sending or storing encrypted data in the cloud will ensure that data is secure. However, it is assumed that the encryption algorithms are strong. Following are a few common security issues faced in the cloud environment across the globe which can be overcome by deploying appropriate security algorithms.

- The physical security is lost in cloud environment as computing resources are shared with other companies. No control exists in the place where resources are running.

- Data integrity is a major issue as it changes in response to transactions like transfer, storage and retrieval. It will be a threat if it's an unauthorized transaction. So, ensuring secured data transfer is a crucial phenomenon which can be achieved by applying these security algorithms.
- Cloud service providers may break the rules by sharing personal information or sensitive data with unintended persons.
- A security issue may arise if encryption-decryption keys are not handled appropriately.

1.4.2 Symmetric Encryption Algorithms

Symmetric key algorithms are primarily meant for bulk encryption of data. This technique is very fast and consists of a big number of possible keys. The best of this set offers superb secrecy. The biggest advantage is the absence of any fast way to decrypt the data without having the same key once data is encrypted with a specific key. These types of algorithms are mainly divided into two categories: block and stream. Block algorithms are responsible for blockwise encryption of data whereas the second category does it byte by byte [14].

The strength of different encryption algorithms may differ as each of them has a specific area of expertise. Some are not very efficient in data protection, they allow decryption of encrypted information without having knowledge of a requisite key, whereas some are very competent at resisting the most obvious attack. The strength of the algorithm depends on several factors, a few of which are listed below:

- Confidentiality of the key
- Effort of guessing the key
- Difficulty in breaking the encryption algorithm
- Existence of other options to decrypt without knowing the key
- The capacity to decrypt an entire encrypted message if the logic of decryption is known for a portion.

Generally, the cryptographic strength is not established; the loopholes of the technique can be analyzed and strength can be assumed accordingly.

Key length is another vital aspect of the algorithms as security is intensely related with the key length. Though short keys compromise the security of the encrypted message, extremely long keys also don't provide the best result due to the complexity present to maintain them. If the length is increased from 80 bits to 128 bits, the effort of guessing the key also significantly increases. So, there is an increased demand for creating longer keys from a marketing perspective.

Table 1.1 below describes a few commonly used symmetric encryption algorithms along with their key length, out of which four are discussed in the subsequent section.

1.4.2.1 Advanced Encryption Standard

The advanced encryption standard (AES) is a symmetric encryption algorithm that supports a block length of 128 bits, capable of using cryptographic keys of 128, 192, and 256 bits, which was developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen. It is able to provide higher level security due to its 256-bit key length and is usually

Table 1.1 Commonly used symmetric encryption algorithms and their key length.

Algorithm	Description	Key Length
Blowfish	Block cipher developed by Schneier	1-448 bits
DES	DES adopted as a U.S. government standard in 1977	56 bits
IDEA	Block cipher developed by Massey and Xuejia	128 bits
MARS	AES finalist developed by IBM	128-256 bits
RC2	Block cipher developed by Rivest	1-2048 bits
RC4	Stream cipher developed by Rivest	1-2048 bits
RC5	Block cipher developed by Rivest and published in 1994	128-256 bits
RC6	AES finalist developed by RSA Labs	128-256 bits
Rijndael	NIST selection for AES, developed by Daemen and Rijmen	128-256 bits
Serpent	AES finalist developed by Anderson, Biham, and Knudsen	128-256 bits
Triple-DES	A three-fold application of the DES algorithm	168 bits
Twofish	AES candidate developed by Schneier	128-256 bits

suggested in several financial organizations for a secured business transaction both online or on corporate network infrastructure to protect the data of clients, staff, partners, and suppliers [15, 16]. The algorithm has the following high level steps. The subsequent Figure 1.2 represents the major steps in the algorithm.

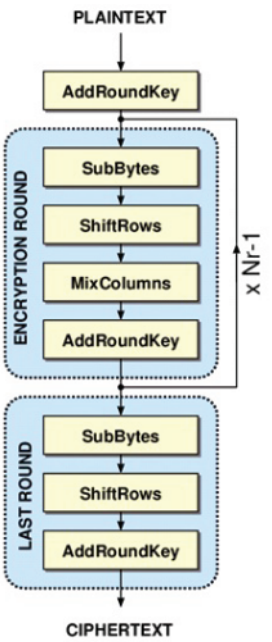

Figure 1.2 Outline of AES algorithm.

Key Expansion: Cipher key derives number of separate round keys from a short key through using Rijndael's key routine. It needs a separate 128-bit round key block for each round and one more. The key schedule builds the required round keys from the basic key.

Initial Round Key Addition: It has the AddRoundKey function where each byte of the state is combined with a block of the round key using bitwise XOR. Here, the initial round key is added with the starting state array.

Encryption Round: The encryption process needs a series of steps to modify the state array. The steps consist of four types of operations. All four of these operations are applied on the current state array and a new version is built. The details of each operation are briefly given below. An iteration of the below steps is called a round. The amount of rounds of the algorithm depends on the key size. Table 1.2 depicts the same.

Table 1.2 Key size and rounds.

Key Size (Bytes)	Block Size (Bytes)	Rounds
16	16	10
24	16	12
32	16	14

- *SubBytes*: A nonlinear substitution step where each byte is replaced with another according to a lookup table. A simple substitution occurs which converts every byte into a different value. AES builds a table containing 256 values required for substitution. For the 16 bytes of the state array, each byte is used as an index into the 256-byte substitution table and the byte is replaced by the value from the substitution table. A new version of the state array is formed because all possible 256 byte values are present in the table. The new version can be restored to its original contents using an inverse substitution table. The contents of the substitution table are not arbitrary; its entries are calculated through a mathematical formula. However, most implementations simply contain the substitution table stored in the memory as a part of the entire design.

- *ShiftRows*: A transposition step where the last three rows of the state are shifted cyclically a certain number of steps. It operates on every row of the state array. A rotation occurs to the right by a specific number of bytes as described below. The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

  – 1st Row: Rotated by 0 bytes (no change)
  – 2nd Row: Rotated by 1 byte
  – 3rd Row: Rotated by 2 bytes
  – 4th Row: Rotated by 3 bytes

- *MixColumns*: A linear mixing operation which operates on the columns of the state, combining the four bytes in each column. Here each column of the state array is processed individually to create a new column. Processing refers matrix multiplication here. The result is another new matrix consisting of 16 new bytes.

- *AddRoundKey*: Now the 16 bytes of the matrix are considered as 128 bits and are XORed with the 128 bits of the round key. The output becomes ciphertext if this is

the last round, otherwise the resulting 128 bits are interpreted as 16 bytes and another similar round started.

Final Round: In the final round, all these steps are performed except the Mix column step to make the algorithm reversible during decryption (SubBytes, ShiftRows, AddRoundKey).

1.4.2.2 3DES

Data encryption standard (DES) is a block cypher where a 56-bit key is used and consists of various operating modes; based on the purpose, they are deployed in various scenarios [17]. There is a limitation of its usage in reality due to its short key length though it's a strong algorithm. 3DES is introduced to fill this gap by using three different keys, having a total key length of 168 bits. This feature has made it extremely secure and it has been adopted by several financial institutions. Currently, two types of triple DES exist which are 3-key Triple DES (3TDES) and 2-key Triple DES (2TDES).

The outline of the 3TDES algorithm is presented in the following Figure 1.3.

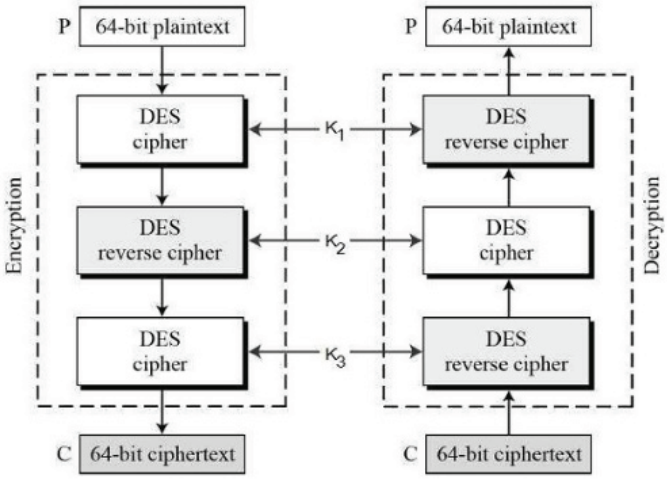

Figure 1.3 Outline of 3-key Triple DES algorithm.

Procedure:

- At first, user generates and distributes a 3TDES key which is the combination of three different keys K_1, K_2 and K_3. So the actual 3TDES key length comes to $3 \times 56 = 168$ bits.
- The plaintext blocks are encrypted by single DES having key K_1.
- The output of the previous step is decrypted using single DES having key K_2.
- In the next step, the output of previous step is encrypted by single DES with key K_3.
- The current output is ciphertext.
- Decrypting the ciphertext is just opposite of the previous one. First, it is decrypted using K_3, encrypted with K_2 and finally decrypted with K_1.

1.4.2.3 Blowfish

Blowfish is a fast, compact, and simple block encryption algorithm which works very efficiently against hackers and cyber-criminals [18]. It supports a wide array of products like secure E-mail encryption tools, backup software, password management tools and TiVo. It allows a variable-length key, up to 448 bits, and is optimized for execution on 32-bit or 64-bit processors. Several famous applications exist which use Blowfish encryption, a few of which include AEDIT (a free Windows word processor incorporating text encryption), Coolfish (an encrypting text editor for Windows) and FoopChat (encrypted chat and advanced file sharing using a client/server architecture).

The algorithm has two major parts: a key expansion part and a data encryption part.

Key Expansion Part or Subkey Generation: Original key is broken by key expansion into a bundle of subkeys:

Algorithm 1.1 Key Expansion Part or Subkey Generation

Begin

1. Key Size is variable but it generates very large subkeys. The key size lies between 32 bits to 448 bits.
2. Concept of P-array consists of 18, 32-bit subkeys.
3. There are 4 S-boxes containing 256 entries of 32 bits.
4. P-array is initialized first then four s boxes with fixed string.
5. P-arrays are XORed with subkeys, i.e., from P_1 to P_{18}.
6. Once the subkeys are generated the encryption process begins.

end

Data Encryption Part: This process involves 16 times iteration of a simple function. Each round contains a key-dependent permutation and key and data substitution.

1.4.2.4 RC6

Rivest Cipher 6 (RC6), an improved version of the basic Rivest Cipher RC algorithm, uses 128-bit block size and supports key sizes of 128, 192 and 256 bits and was developed by Ron Rivest, Matt Robshaw, Ray Sidney, and Yiqun Lisa Yin to meet the requirements of an advanced encryption standard (AES) competition. It may be parameterized to support a wide variety of word lengths, key sizes and number of rounds. Considering the structure, RC6 and RC5 are similar; they use data-dependent rotations, modular addition, and XOR operations. Being a secure, compact and simple block cipher, good performance and flexibility are offered [13, 19].

New features of RC6 include using four working registers instead of two and the presence of integer multiplication as an additional primitive operation. Using multiplication greatly increases the diffusion achieved per round, allowing for greater security, fewer rounds and increased throughput.

1.4.3 Asymmetric Encryption Algorithms

Asymmetric encryption refers to public key cryptography which is a new method compared to symmetric encryption. Here two keys are used for a plain text encryption. Secret keys

are exchanged over the network which ensures that malicious persons cannot misuse the keys. The message can be decrypted by a person having the secret key and that is the reason for using two related keys for boosting security. A public key is made freely accessible to everybody who wants to send a message whereas a private key is kept secret. A message encrypted by a public key can only be decrypted by the private key. However, a message encrypted by a private key can be decrypted by a public key. A public key need not be secured as it's available to everybody; hence, it can be passed through the internet. The technique offers better security than symmetric encryption during message transmission as it uses two keys, though it takes a longer time than the other one. A few well-known techniques are described below which follow asymmetric encryption principles [16].

1.4.3.1 RSA

The RSA[1] algorithm was developed by Ron Rivest, Adi Shamir and Len Adleman in 1977 and was named after the inventors [14, 19]. This is the most widely used public key cryptography algorithm across the globe. It has the capability to encrypt a message without exchanging the secret key separately. It can be applied in both public key encryption and digital signatures. Security level depends on the difficulty of factoring large integers. For example, an encrypted message can be sent to *B* from *A* without any previous communication of secret keys. *A* applies *B*'s public key for message encryption whereas *B* uses the private key for decryption. Similarly, the technique is applicable to sign a message where it can be signed by *A* with its private key and can be verified by *B* using *A*'s public key.

The basic version of the algorithm is depicted as follows:

Algorithm 1.2 RSA

Begin

1. Produce *p* and *q*, two large random primes, having approximately equal size, and their product $n = p \times q$, which should be the required bit length.
2. Calculate *n* and φ such that $n = p \times q$ and $\varphi = (p-1) \times (q-1)$.
3. Select an integer *e* where $1 < e < \varphi$ and $gcd(e, \varphi) = 1$.
4. Calculate the secret exponent *d* where $1 < d < \varphi$ such that $ed \equiv 1 \bmod \varphi$.
5. The public key is (n, e) and private key is (d, p, q).

end

Note:

- *n* is commonly called the modulus.
- *e* is called the public exponent/encryption exponent.
- *d* is called the secret exponent/encryption exponent.

After generation of the public key and private keys from the above steps, Encryption, Decryption, Digital signing and Signature verification occurs in the following ways.

Similarly, it works for the domain of Digital Signing.

[1]https://www.rsa.com/

Algorithm 1.3 RSA Encryption (Performed by Sender *A*)

Begin

1. Obtains the receiver *B*'s public key (n, e).
2. Designate the plain text message as a positive integer *m* where $1 < m < n$.
3. Calculates the cipher text $c = m^e \bmod n$.
4. Transmit the cipher text *c* to *B*.

end

Algorithm 1.4 RSA Decryption (Performed by Receiver *B*)

Begin

1. Applies his private key (n, d) to calculate $m = c^d \bmod n$.
2. Plain text is pulled out from the message representative *m*.

end

Algorithm 1.5 RSA Digital Signing (Performed by Sender *A*)

Begin

1. A message digest, the information needed to be sent, is built.
2. The digest is represented as an integer *m*, which lies between 1 and $n - 1$.
3. The private key (n, d) is used to generate the signature $s = m^d \bmod n$.
4. The signature *s* is sent to the receiver *B*.

end

Algorithm 1.6 RSA Digital Verification (Performed by Receiver B)

Begin

1. The public key (n, e) of sender *A* is used to compute integer $v = s^e \bmod n$.
2. The message digest *H* is extracted from the integer.
3. The message digest H' of the information that has been signed is calculated separately.
4. Check whether $H = H'$.
5. The signature is valid if $H = H'$ is true.

end

Summary of RSA:

1. $n = p \times q$ where *p, q* are distinct primes
2. $\varphi = (p-1)(q-1)$
3. $e < n$ so that $gcd(e, \varphi) = 1$
4. $d = e^{-1} \bmod \varphi$
5. $c = m^e \bmod n, 1 < m < n$
6. $m = c^d \bmod n$

1.4.3.2 Diffie-Hellman Key Exchange

The Diffie-Hellman key exchange protocol was invented in 1976, and was the first technique to establish a shared secret over an unsecured communication channel. The concept of key exchange was one of the first problems addressed by cryptography. The main idea is to use a key which can be used by two parties for encryption in such a way that an eavesdropper cannot get the key [20]. The outline of the algorithm is presented below.

Assumption:

- Here, four variables are considered to keep it simple and for practical implementation. Variables include prime *p*, *g* (a primitive root of *p*), two private values *a* and *b*. Both *p* and *g* are publicly available numbers.
- Users (consider Alice and Bob) choose private values a and *b*. A key is also generated and exchanged publicly. Another person receives the key and generates a secret key so that they both have the common secret key to encrypt.

Algorithm 1.7 Diffie-Hellman Key Exchange

Begin

1. Alice and Bob both decide on *a* prime number *p* and *a* base *g*.
2. Alice selects a secret number *a*, computes ($x = g^a \bmod p$) and sends Bob *x*.
3. Bob selects a secret number *b*, computes ($y = g^b \bmod p$) and sends Alice *y*.
4. Alice calculates ($y^a \bmod p$).
5. Bob calculates ($y^a \bmod p$).
6. By the laws of algebra, Alice's key is the same as Bob's key.

end

Both Bob and Alice can use this number as their key. However, *p* and *g* need not be protected.

1.4.4 Security Enhancement in Cloud Using Encryption Algorithms: Observations

Research work has been carried out to find the utility of deploying the security algorithms in cloud. A few of these observations are listed below [19, 21].

- Data security in cloud-based applications can be enhanced to a great extent by applying RSA and AES algorithms.
- Private key determination is not feasible for hackers (*even if they have the public keys generated*) if RSA uses 1024-bit keys and AES uses 128-bit keys.
- In a scenario where the user forgets to log out from the cloud web portal after using it and an attacker breaks into the user system, it will not be possible to access or download data without entering the private key in the system.
- However, if the attacker is successful in breaking into the user's system and guesses the private key, it may be possible to download the encrypted data but accessing the original data still remains a hurdle.

1.5 Comparison of Encryption Algorithms

The following Table 1.3 presents a comparison of the encryption algorithms discussed above based on various parameters.

Table 1.3 Comparison of encryption algorithms.

Algorithm / *Parameter*	3DES	AES	Blowfish	RSA	Diffie-Hellman
Encryption Technique	Symmetric key	Symmetric key	Symmetric key	Asymmetric key	Asymmetric key
Used Keys	Same key in Encryption and Decryption	Same key in Encryption and Decryption	Same key in Encryption and Decryption	Different key in Encryption and Decryption	Key Exchange
Throughput	Lower than DES	Lower than blowfish	Very High	High	Low
Key Lengths	112 - 168 bits	128,192 or 256 bits	32 - 448 bits	>1024 bits	Key Exchange Management
Rounds	48	10,12,14	16	1	56
Security Against	Brute Force, Chosen-plain text, Known plain text	Chosen-plain text, Known plain text	Dictionary Attacks	Timing Attacks	EavesDropping
Modification	The key size is increased from 56 to 168 bits	128, 192 or 256. Its structure was flexible to multiples of 64	Key length in blowfish should be multiples of 32	Key length in RSA algorithm can be 256 ,512,1024,2048, 4096 bit	No modification in key length
Created by	IBM	Vincent Rijmen , Joan daeman	Bruce Schiener	Ron Rivest,Shamir & leonard Adleman	Whitfield diffie, Martin Hellman
Year	1978	1978	1993	1978	2002
Algorithm Structure	Feistal Structure	Feistal Structure	Feistal Structure	Feistal Structure	Tree Based
Cloud Compatibility	Yes	Yes	Yes	Yes	Yes
Algorithm used in Cloud	Not used in Cloud	Google Drive, OneDrive, Dropbox	Mozy Backup, Foopchat, GigaTribe	Amazon web Services, RSAWeb	CurveCP
Application	Microsoft OneNote,Outlook 2007	Password Manager	IDS Server,Sql Server 2000	Online Credit Card Security System,RSA Signature Verification	Protocols like SSL,SSH,IPSec

1.6 Performance Analysis of Encryption Algorithms in Cloud

The encryption algorithms discussed above are analyzed in terms of mean processing time in local system as well as cloud network. The mean time refers to the difference between the start and end time of encryption taken by a particular algorithm, which is calculated in milliseconds. The following Table 1.4 presents the comparison details. It is evident from the table that each algorithm takes much less time in cloud compared to its non-cloud environment, which again proves the suitability of implementing these algorithms in cloud.

Table 1.4 Comparison performance of encryption algorithms.

Input	AES	AES Cloud	DES	DES Cloud	Blowfish	Blowfish Cloud
10 KB	11.5	1.5	7.5	2	4	2
13 KB	14.7	2	10	2.5	4.7	2
39 KB	21	3	3.15	6.5	8.25	2.75
56 KB	24.5	3.75	50.25	9.25	15.7	3

1.7 Conclusion

Moving to cloud is the ultimate objective for most of the industries and organizations as it involves multiple benefits like using software which is not present in the computer or accessing data from anywhere in the world. However, security, privacy and data theft are the challenges of this platform, which demand the appropriate deployment of security algorithms in cloud to ensure end-user security. The cloud providers generally use a set of tools and techniques to ascertain this. But implementing the encryption algorithms discussed in this chapter definitely makes the platform more secure and reliable for its users. As a result, the usage of cloud computing will be enhanced and will become more popular among its users.

REFERENCES

1. Lee, G. (2010). *Cloud Computing: Principles, Systems and Applications*/Nick Antonopoulos, Lee Gillam. L.: Springer.
2. Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.
3. Hfer, C. N., & Karagiannis, G. (2011). Cloud computing services: taxonomy and comparison. *Journal of Internet Services and Applications*, 2(2), 81-94.
4. Kartit, Z., Azougaghe, A., Idrissi, H. K., El Marraki, M., Hedabou, M., Belkasmi, M., & Kartit, A. (2016). Applying Encryption Algorithm for Data Security in Cloud Storage. In *Advances in Ubiquitous Networking* (pp. 141-154). Springer, Singapore.
5. Bhardwaj, A., Subrahmanyam, G. V. B., Avasthi, V., & Sastry, H. (2016). Security algorithms for cloud computing. *Procedia Computer Science*, 85, 535-542.
6. Iyer, S. S., Dand, H., & Patil, R. (2017). *An Algorithm for Encrypted Cloud Communication*.
7. Conner, W., Iyengar, A., Mikalsen, T., Rouvellou, I., & Nahrstedt, K. (2009, April). A trust management framework for service-oriented environments. In Proceedings of the 18th international conference on World wide web (pp. 891-900). ACM.
8. Friedman, A. A., & West, D. M. (2010). Privacy and security in cloud computing. Center for Technology Innovation at Brookings.
9. Ristenpart, T., Tromer, E., Shacham, H., & Savage, S. (2009, November). Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In Proceedings of the 16th ACM conference on Computer and communications security (pp. 199-212). ACM.
10. Yan, L., Rong, C., & Zhao, G. (2009, December). Strengthen cloud computing security with federal identity management using hierarchical identity-based cryptography. In IEEE International Conference on Cloud Computing (pp. 167-177). Springer, Berlin, Heidelberg.

11. Ukil, A., De Sarkar, A., Jana, D., & Wyld, D. C. (2013). Security policy enforcement in cloud infrastructure. In ICCSEA, SPPR, CSIA, WimoA-2013 (pp. 01-09).

12. Le, D. N., Kumar, R., Nguyen, G. N., & Chatterjee, J. M. (2018). *Cloud Computing and Virtualization*. John Wiley & Sons.

13. Le, D. N., Kumar, R., Mishra, B. K., Chatterjee, J. M., & Khari, M. (Eds.). (2019). *Cyber Security in Parallel and Distributed Computing: Concepts, Techniques, Applications and Case Studies*. John Wiley & Sons.

14. Shaik, K., Rao, N., & Venkat, T. (2017). Implementation of Encryption Algorithm for Data Security in Cloud Computing. *International Journal of Advanced Research in Computer Science*, 8(3).

15. Shinde, M. R., & Taur, R. D. (2015). Encryption Algorithm for Data Security and Privacy in Cloud Storage. *American Journal of Computer Science and Engineering Science.*

16. e-tutorials accessed at http://etutorials.org/Linux+systems/unix+internet+security/Part+II+ Security+Building+Blocks/Chapter+7.+Cryptography+Basics/7.2+Symmetric+Key+Algorithms/

17. Jain, N., & Kaur, G. (2012). Implementing DES algorithm in cloud for data security. VSRD *International Journal of Computer Science & Information Technology*, 2(4), 316-321.

18. Iyer, S. S., Dand, H., & Patil, R. (2017). *An Algorithm for Encrypted Cloud Communication*.

19. Mewada, S., Shrivastava, A., Sharma, P., Purohit, N., & Gautam, S. S. (2015). Performance Analysis of Encryption Algorithm in Cloud Computing. *International Journal of Computer Sciences and Engineering*, 3, 83-89.

20. Ahmed, M., Sanjabi, B., Aldiaz, D., Rezaei, A., & Omotunde, H. (2012). Diffie-Hellman and its application in security protocols. *International Journal of Engineering Science and Innovative Technology* (IJESIT), 1, 69-73.

21. Le, D. N., Seth, B., & Dalal, S. (2018). A Hybrid Approach of Secret Sharing with Fragmentation and Encryption in Cloud Environment for Securing Outsourced Medical Database: A Revolutionary Approach. *Journal of Cyber Security and Mobility*, 7(4), 379-408.