

# **Framework and Algorithms to Improve Security Mechanism in Cloud**

Project Report submitted in partial fulfillment of the requirement for the  
degree of

Master of Technology  
in  
**Computer Science & Engineering**

under the Supervision of

*Dr. P. K. Gupta*

Co-Supervision of

*Prof. Dr. S. P. Ghrera*

By

*Gunjan Gugnani (132223)*



May - 2015

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,  
Waknaghat, Solan – 173234, Himachal Pradesh

## **Certificate**

This is to certify that project report entitled “**Framework and Algorithms to Improve Security Mechanism in Cloud**”, submitted by **Gunjan Gugnani (132223)** in partial fulfillment for the award of degree of Master of Technology in Computer Science and Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

**Date**

**Signature**

**Supervisor’s Name**

**Dr. P. K. Gupta**

**(Assistant Professor)**

**Signature**

**Co-Supervisor’s Name**

**Prof. Dr. S. P. Ghrera**

**(H.O.D.- C.S.E. Deptt. )**

## Acknowledgement

We would like to take this opportunity to express our sincere indebtedness and sense of gratitude to all those who have contributed greatly towards the successful partial completion of my thesis “**Framework and Algorithms to Improve Security Mechanism in Cloud**” .

It would not have been possible to see through the undertaken thesis without the guidance and constant support of our **Supervisor Dr. P. K. Gupta and Co-Supervisor Prof. Dr. S. P. Ghreera**. For his coherent guidance I feel fortunate to be taught by him, who gave me his unwavering support. I owe my heartiest thanks to **Prof. Dr. RMK Sinha (Dean.-CSE Department)** who has always inspired confidence in us to take initiative.

As a final note, we are grateful to CSE and IT Department of Jaypee University of Information and Technology ,who inspired us to undertake difficult tasks by their strength of understanding our calibre and our requirements and taught us to work with patience and provided constant encouragement to successfully complete the project.

Date:

Name of Student

Gunjan Gugnani

# Table of Contents

<i>List of Figures</i> .....	V
<i>Lists of Tables</i> .....	Vii
<i>Abbreviations</i> .....	Viii
<i>Abstract</i> .....	X
<b>Chapter 1-INTRODUCTION</b> .....	<b>1</b>
1.1 Cloud Computing.....	2
1.2 Essential Characteristics.....	3
1.3 Benefits of Cloud.....	6
1.4 Cloud Security.....	7
1.5 Cloud Security Controls.....	8
1.6 Need of Security.....	9
1.7 Threats in Cloud Computing.....	10
1.8 Attacks on Cloud.....	12
1.9 Security Services.....	14
1.10 Problem Statement.....	16
1.11 Thesis Organization.....	16
<b>Chapter 2-LITERATURE REVIEW</b> .....	<b>17</b>
2.1 Cloud Security.....	18
2.2 XML and Attacks.....	20
2.3 Embedding Confidentiality.....	21
2.4 Encryption Techniques.....	24
<b>Chapter 3-PROPOSED WORK</b> .....	<b>27</b>
3.1 XML Encryption.....	28
3.2 DNA Encryption/Decryption.....	29
3.3 DNA Algorithm.....	31
3.4 Comparison.....	32
3.6 XPath Injection Attack.....	36
<b>Chapter 4 – IMPLEMENTATION and RESULTS</b> .....	<b>40</b>
4.1 Introduction to CloudSim Simulation.....	42
4.2 Results.....	44

4.3 Analysis of Proposed Approach.....	45
4.4 DES Analysis.....	46
4.5 AES Analysis.....	48
4.6 RSA Analysis.....	49
4.7 Comparison with Traditional Scheme.....	50
4.8 Comparison with Every Field Encryption.....	52
4.9 Attack.....	53
<b>Chapter 5 – CONCLUSION.....</b>	<b>54</b>
5.1 Conclusion.....	55
5.2 Future Work.....	55
<b><i>List of Publications</i>.....</b>	<b>56</b>
<b><i>References</i>.....</b>	<b>57</b>
<b><i>Appendix</i>.....</b>	<b>61</b>

## List of Figures

<b>S.No</b>	<b>Title</b>	<b>Page No.</b>
1	Different Ways to Communicate with Cloud	3
2	Characteristics of Cloud	5
3	Benefits of Cloud	5
4	Different Types of Attacks	14
5	Data Obfuscation	23
6	SCoRiM Framework	23
7	Asymmetric DNA Scheme	26
8	DNA Encryption	30
9	DNA Decryption	30
10	DES Encryption	33
11	AES Encryption, Decryption	34
12	RSA Encryption, Decryption	35
13	Output of Cloud	43
14	XML File	44
15	GUI for Encryption	44
16	Number of Bits vs. Average DNA Encryption Time	45
17	Output DNA encryption	46
18	Output DES	47
19	Average Encryption Time of DES vs. No. of Bits	47
20	Output of AES	48
21	Average Encryption Time of AES vs. No. of Bits	49
22	Output of RSA	49
23	Average Encryption time of RSA vs. No. of Bits	50

24	Comparison between DNA, AES, DES and RSA	51
25	Average Decryption Time for Different Algorithms	52
26	GUI for Attack	53
27	Fetch details via Attack	53

## List of Tables

<b>S.No.</b>	<b>Title</b>	<b>Page No.</b>
1	Attacks with Example	15
2	Comparison between Traditional and DNA Cryptography	24
3	DNA Sample Output	32
4	Attack Before Encyption	38
5	Attack After Encryption	38



# Abbreviations

1. DNA – Deoxyribonucleic acid
2. XML – Extensible Markup Language
3. SLA – Service Legal Agreement
4. VM – Virtual Machine
5. DoS – Denial of Service
6. DDoS – Distributed Denial of Service
7. SSL – Secure Socket Layer
8. CSP – Customer Service Provider
9. OTP – One Time Pad
10. DES – Data Encryption Standards
11. DSA- Digital Signature Standards
12. AES – Advanced Encryption Standards
13. RSA – Rivest Shamir Adleman
14. NIST - National Institute of Standards and Technology
15. SaaS- Software as a Service
16. IaaS - Infrastructure as a Service
17. PaaS - Platform as a Service
18. OS – Operating System
19. PC – Personal Computer
20. OSSTMM- Open Source Testing Methodology Manual
21. IDS- Intrusion Detection System
22. IPS- Intrusion Prevention System
23. SQL- Sequential Query Language
24. WSDL- Web Services Description Language
25. Key Management Interoperability Protocol
26. PDI- Provable Data Integrity (PDI)
27. IA- Information Accountability
28. SOAP- Simple Object Access Protocol
29. ATM- Automatic Teller Machine
30. ECC – Elliptic Curve Cryptography
31. IDE – Integrated Development Environment

32. RAM- Random Access Memory

33. GUI – Graphical User Interface

## Abstract

Hottest buzzword of this decade “Cloud Computing” - which has been considered as one of the potential solutions to our increasing demand for accessing, processing, storing and using provisioned resources over the internet. However with so many boons, it comes along with some curse as security and trust issues. There are many security issues within cloud like improper information disclosure and dissemination. So as to provide confidentiality to cloud users this work proposes an XML encryption technique. The encryption technique used is DNA encryption by using some interesting features of some DNA sequencing. It assures not to disclose the user’s confidential information. DNA encryption technique is compared with other encryption schemes in literature as with DES, AES and RSA. During the experiments of selectively encrypting an XML file, it was observed that encryption time as compared to AES, DES and RSA is significantly very low also in security point of view, the probability to decrypt a DNA encrypted text is also very low. Moreover a comparison is made between the average encryption time of selected text with the average encryption time of full content of file. During the experiments it was found that there lies a huge difference in average encryption between full and selective encryption average time, the difference was not merely in the time but was also that the difference of length of plaintext and cipher text. The cipher text after encryption is very large. Hence if we choose selective encryption then it not only maintains the confidentiality but also improves performance by saving encryption time and space used for encryption. In this work an XPath attack on XML file is also demonstrated and the experiments show that by encrypting the XML data and storing the data on XML in encrypted form can prevent the information disclosure, hence provide the confidentiality to the user.

# Chapter 1

## **Introduction**

*Cloud Computing*

*Characteristics*

*Cloud Security Controls*

*Need of security*

*Threats and Attacks*

*Security Mechanism*

# CHAPTER - 1

## INTRODUCTION

The cloud computing framework offers dynamically scalable resources provisioned as a service over the internet. There are many economic benefits as it promises to reduce the capital expenditure and operational expenditure. But there are still some challenges left to focus on. Out of those challenges- security and trust are the foremost issues. The data that is being transferred between user & cloud and also the data residing on the cloud needed to get secured from different threats and attackers.

So our work purposes a proficient approach to embed confidentiality in the system and moreover makes the system more robust by preventing the system from various attacks.

### 1.1 Cloud Computing

According to NIST (National Institute of Standards and Technology) Cloud computing is an internet based model for enabling handy, omnipresent and on demand network access to a common pond of configurable computing capabilities that can be quickly allocated and freed with negligible administrative effort or by least interaction with cloud service provider [1].

In other words Cloud computing is Internet based computing where shared resources, software, services and information are provided to computers, mobile phones and other devices on demand(Figure1).Cloud computing is a way of computing in which dynamically scalable and often virtualized resources are supplied as service over the Internet on demand.

The cloud computing model is formed by five vital characteristics, majorly there are three service models and generally four deployment models. Three Cloud service models are - Software as a Service (SaaS), second is Infrastructure as a Service (IaaS) and the third one is Platform as a Service (PaaS). Cloud computing model has four

deployment models - one is Private cloud, second is Public cloud, third is hybrid cloud and the last is Community cloud.

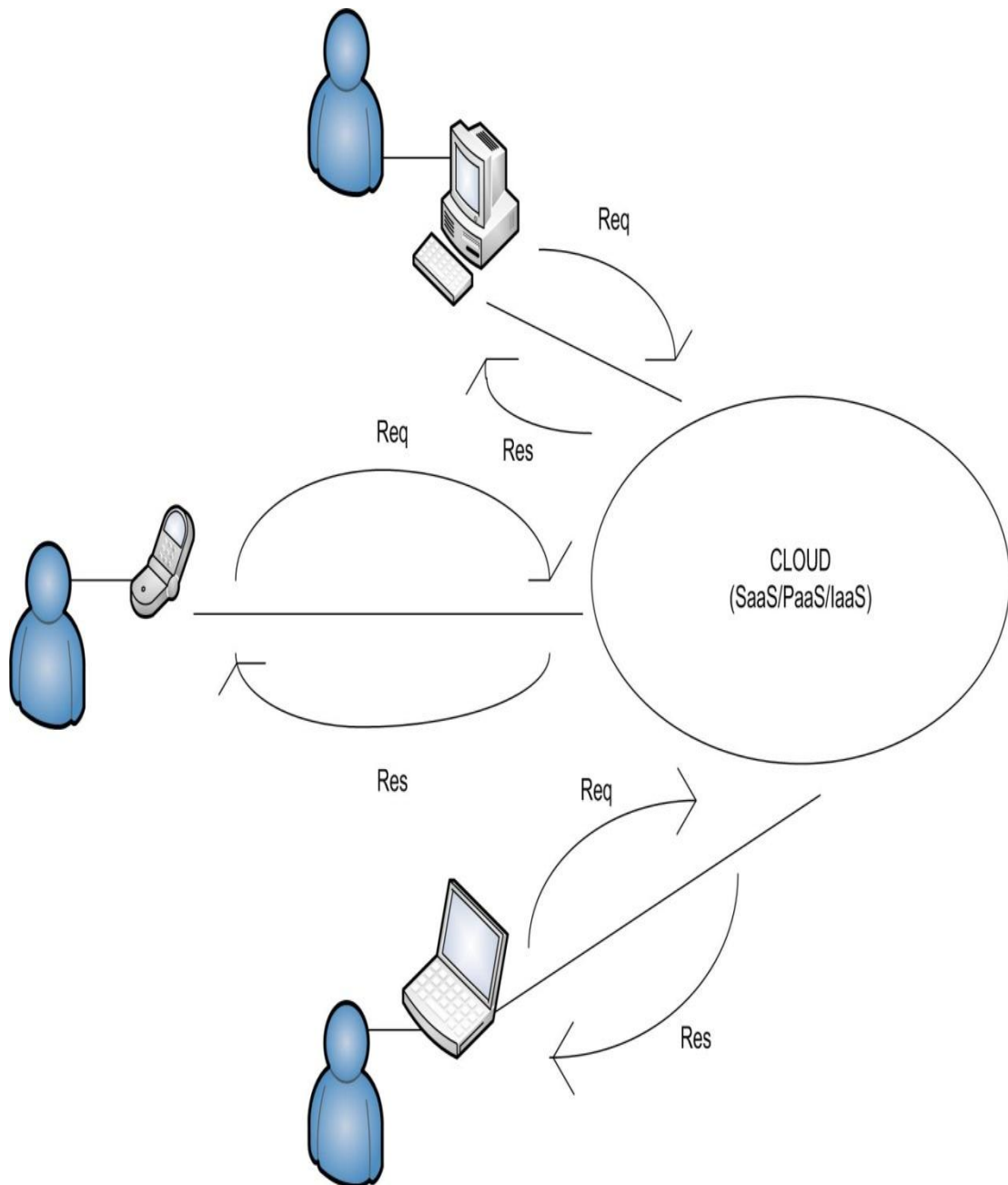


Figure1: Different Ways to Communicate with Cloud.

## 1.2Essential Characteristics:

A cloud must possess some essential characteristics. Every cloud projected to have these five characteristics (Figure2).

### **a) On-demand self-service**

A cloud client can unilaterally provision computing resources for e.g. network storage and server processing time as needed automatically without requiring one's individual interaction with each cloud service provider. Example such as a Web Portal and Management interface [13] in this one can himself ask for the service and will get the requested service unilaterally. Provisioning and de-provisioning of OS services and associated resources take place automatically at the provider end.

### **b) Broad network access**

Cloud resources are available over the network and can be used through some standard methods that support access by heterogeneous thick or thin customer platforms (for instance- mobile phones, laptops, personal computers and tablets).

### **c) Resource pooling**

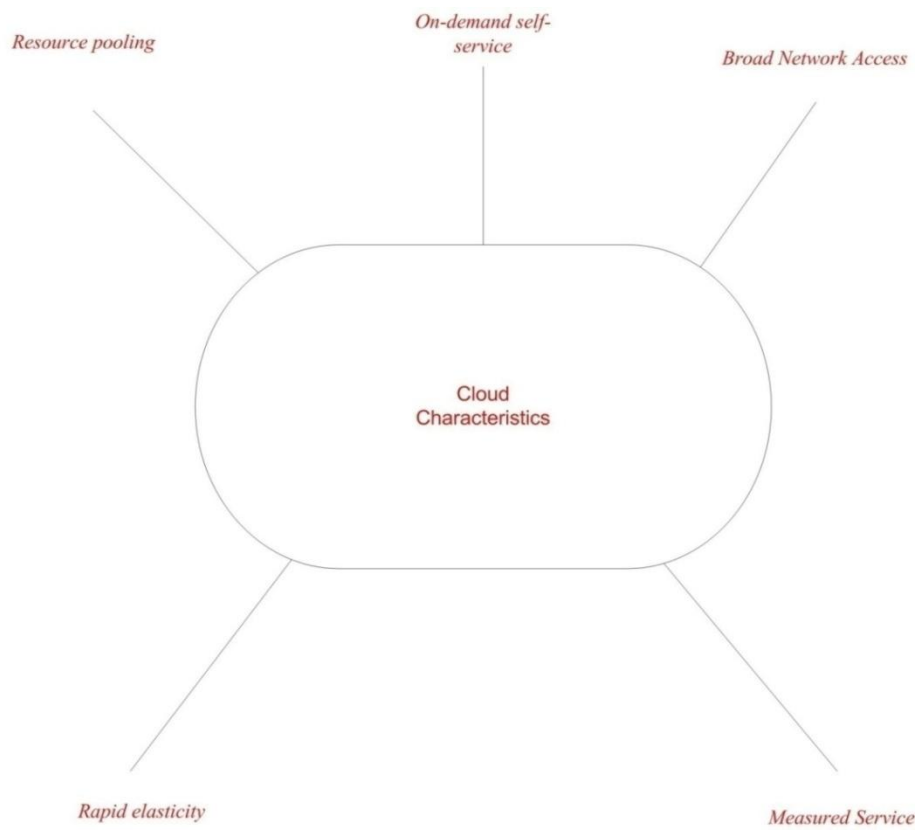
The service provider's computing resources are collected to provide multiple consumers using a multi tenant model with diverse physical and virtual resources dynamically assigned and reassigned according to customer order. There is an independence of place and location to the cloud client that he generally does not handle or doesn't know where the provided resources are located but there is a possibility that at a much higher level of abstraction he is able to specify the location of resources (e.g., countries, states, or datacenters). Examples of resources include storage, memory, processing and network bandwidth.

### **d) Rapid elasticity**

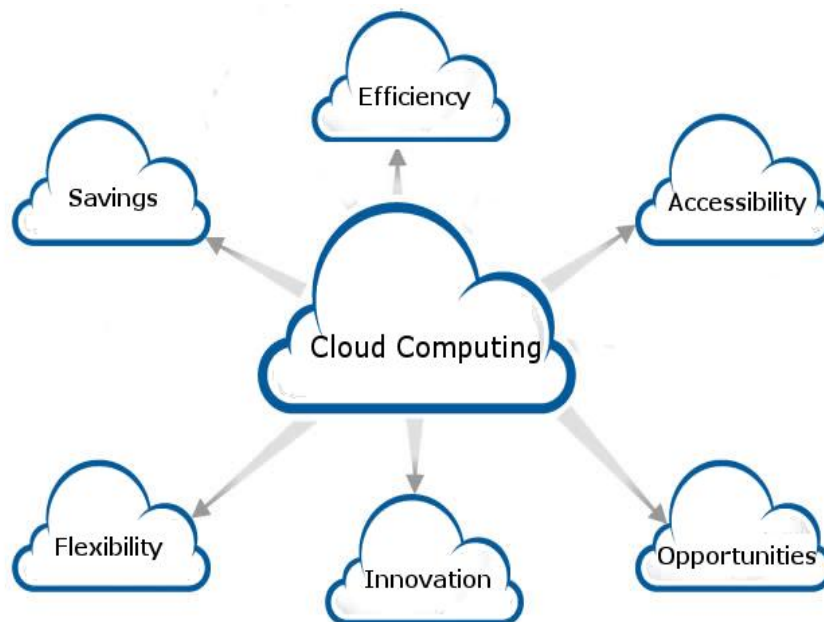
Resources can easily be provisioned and freed automatically to scale quickly outward and inward adequate with demand. Cloud customers often see resources as unlimitedly available for provisioning and can be appropriated in any quantity at any time.

### **e) 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. such as storage, bandwidth, processing and active user accounts). Capability utilization can be examined, measured, handled and reported and also provides transparency to both the provider and the cloud customer of the utilized resource.



**Figure 2: Characteristics of Cloud**



**Figure3: Benefits of Cloud**



## **1.3 Benefits of Cloud**

In today's business scenario there are many benefits of cloud. The benefits are as follows(Figure3):

### **a) Cost savings**

Companies can use operational expenditure and reduce their capital expenditure in order to increase their computing capabilities.

### **b) Scalability and flexibility**

Companies can start with small deployment and with fair speed they can grow to large deployment and then can even scale back if required. The flexibility feature of cloud computing allows the companies to use extra resources at peak times so as to satisfy the customers.

### **c) Reliability**

Services that make use of multiple redundant sites can support disaster recovery and business continuity.

### **d) Maintenance**

The maintenance of system is done by the cloud service providers and access is through APIs which do not require application installations onto PCs, thus further reducing the maintenance requirements.

### **e) User-centric interfaces**

Cloud computing uses the concept of utility computing in which it becomes easy for consumers to obtain and employ the platforms in computing Clouds.

### **f) On-demand service provisioning**

The resources and services are made available to the users according to their need.

### **g) QoS guaranteed**

Cloud computing guarantees that quality of service would be rendered to its users.eg: size of memory, CPU speed etc.

### **h) Autonomous System**

Cloud computing is an autonomous system and managed transparently to consumers.

### **i) Pay-as-you-go**

This means that payment made by the user is according to the consumption of the resource.

### **j) Energy efficiency**

The cloud reduces the consumption of unused resources thus making this technology an energy efficient technology.

### **k) Multi-tenancy**

Services are owned by multiple providers in a cloud environment that are located in a single data center.

### **l) Service oriented**

Cloud computing follows a service driven model where each PaaS, SaaS, IaaS providers provide service according to the Service Level Agreement (SLA) that is negotiated with its customers.

## **1.4 Cloud Security**

Cloud computing model leaves the clients vulnerable to different types of attacks and threats. Due to this client may suffer from a heavy loss of any confidential data or may lose any confidential information. An attacker may eavesdrop the conversation between two clients on cloud or may eavesdrop the communication between client and cloud. He may harm the client in many ways. So there is a great need to protect the clients from these attacks.

So first understanding the Security-In literature Security is the degree of protection from, resistance to harm. It applies to any valuable and vulnerable asset. According to OSSTMM 3, security offers a method of safeguard where a partition is formed between the valuable assets and the threats. These partitions are usually called controls and sometimes comprise variation to the asset or the threat.

**Cloud Security** refers to a broad variety of technologies, policies, mechanisms, frameworks and controls deployed to protect data, applications and the associated infrastructure of cloud computing model.

Some areas that require focus to embed security

- To safeguard all end cloud - user activities, actions regardless of device
- To protect cloud, database and data centers
- To facilitate superior cyber security against various attacks.

## **1.5 Cloud Security Controls**

Cloud security mechanism is effective only when the correct defensive implementations are in place. Well-organized cloud security mechanism should recognize and address the issues that will arise with security management. These defenses and controls are set in position to defend any flaws in the system and decrease the consequence of an attack. There are different types of controls following cloud security architecture. They can generally be found in one of the following categories [15]:

### **a) Deterrent controls**

The Deterrent controls are anticipated to decrease attacks on a cloud system. Similarly a warning sign on an asset, these deterrent controls typically diminish the effect of threat by notifying the attackers that there will be poor consequences for them if they continue further or move forward in that particular direction .

### **b) Preventive controls**

These controls toughen the system against threats and attacks generally by plummeting if not truly eradicating vulnerabilities. Well-built authentication of cloud customer makes it less possible that unauthorized customer can access cloud systems and more possible that cloud customers are optimistically identified.

### **c) Detective controls**

These controls are proposed to sense and respond appropriately to any accidents that occur. When an attack occurs, this control will sign the anticipatory or

remedial controls to tackle the issue. System and network sanctuary monitoring, intrusion detection and prevention arrangements, are typically engaged to sense attacks on cloud systems

#### **d) Corrective controls**

These controls decrease the result of a threat, usually by restricting the harm. These controls generally come into effect during or after an attack have occurred. Re-establishing system backup so as to reconstruct a cooperated system can be seen as an paradigm of a corrective control.

### **1.6 Need of Security**

In past three decades, there is a huge change in the world of computation from centralized - client-server to distributed systems and now we are going back to the virtual centralization [3]. Location of customer or company's data and processes built this divergence in the area of computation. On one side, a person has full control over his data and processes on his/her own computer. On the other side, we have the cloud computing model where the services and data storage, maintenance, processing is supplied and provided by some vendor which leaves the client/customer uninformed of where his processes are running or where his data is being stored. So, logically the client has no control over his or her data. For communication cloud uses the internet. If we concentrate on the security of data residing on, the cloud provider has to endow with some guarantee in service level agreements (SLA) to persuade the client on security concerns.

If we take public cloud then we can say its environment is extremely complex [2] as compared to a conventional data center situation. If we see the standard scenario of Cloud computing, an organization capitulate straight control over key aspects of security, giving a considerable level of confidence onto the Cloud supplier. Patrons may not be conscious about the detailed security-incidents, accidents, threats, vulnerability or any malware reports.

## **1.7 Threats in Cloud Computing**

Cloud security alliance in presented a primary draft for threats relevant to the security architecture of Cloud services. We discuss here some potential threats relevant to Cloud and relevant mitigation directives.

### **a) Malicious insiders**

Majority of the companies conceal their strategies about the height of access to their staff. Though, via superior level of access, a member of staff can grow access to top secret data and services. As there is deficiency in transparency of Cloud provider's policies, processes and procedures, some insiders can frequently have the privilege to access the client's data. Malicious insider's (employee) actions are often evade by a firewall or Infringement discovery system considering it to be an authorized action. Though, a trusted member of staff may also convert into an opponent. In these kinds of scenarios, insiders can source a significant effect on Cloud services. Lets take an instance-here malicious insiders can access top secret data and put on control over the Cloud services without any jeopardy of revealing his identity. These kinds of threats may be applicable to any cloud service SaaS, PaaS and IaaS. So as to avoid these kinds of risks there is the need of more transparency in security and management process together with compliance reporting and breach notification.

This is amplified in the cloud via the meeting of IT services and client over a single management domain which is united within a general transparency deficiency into the service supplier process and procedure [10].

### **b) Shared technology issues/multi-tenancy nature**

Virtualization in multi-tenant architecture is used to provideshared on-demand services. Different users who have access to the virtual machine may use the same shared application. Though, as mentioned above, via some attacks and threats some malicious entities can gain access and control of the lawful users' virtual machine. In multi-tenant architecture through shared resources IaaS services are delivered which sometimes are not designed to give sufficiently strong isolation. Giving permission to one tenant to interfere in the other can cause serious affect on the cloud architecture which can affect its regular operations. Generally these types of threats have an effect

onIaaS. Transparency in SLA for patching, well formed authentication system and access control mechanisms to administrative tasks are some of the solutions to resolve this issue.

### **c) Data loss and leakage**

Information can be given and taken in many ways. This can incorporate data insertion, data compromise, deletion or modification. As the cloud is shared and dynamic in nature so these threats could prove to be a foremost concern leading to data theft. Instances of these threats are deficiency in authentication and authorization systems, weak encryption algorithms, weak keys, erratic data center, and lack of disaster recovery. This threat can affect to SaaS, PaaS, and IaaS services of cloud. Some of the solutions are safety of Interface, data integrity, use of strong and robust key algorithms, secure storage for used keys, data backup and preservation policies.

### **d) Service hijacking**

Service hijacking is a very serious threat in this the hijacker may forward the cloud customer to an illegal website. For attackers service instances and User accounts can serve as a new base for attack. Some phishing attacks, frauds, exploitation of software vulnerabilities, reused credentials and passwords may pose service or account hijacking. This threat can have great impact on IaaS, PaaS, and SaaS. Here are some of the solutions to resolve this threat which include safety policies, strong authentication and activity monitoring.

### **e) Identity theft**

Identity theft is a kind of scam in which one act as if to be someone else so as to access resources or obtain credit and other benefits. The casualty (of identity theft) can undergo undesirable consequences and losses and held responsible for the criminal's activity. Some of the security perils are phishing attacks, weak password recovery workflows, key loggers etc. This threat can have significant impact on SaaS, PaaS and IaaS and some of the solution includes the use of powerful encryption, authentication and authorization mechanisms.

## 1.8 Attacks on Cloud

There are number of security issues associated with cloud computing but they fall in two categories: security issues faced by cloud provider and security issues faced by their customers. Here are the some risks or attacks on cloud(Figure4).

### a) Zombie Attack

Over the Internet, invaders attempt to flood the victim by sending requests from innocent hosts in the network. These types of hosts are called *zombies*. In the Cloud, the requests for Virtual Machines (VMs) are accessible by each user through the Internet. An attacker can overflow the huge number of requests via *zombies*. This kind of attack disrupts the normal performance of Cloud disturbing the availability of Cloud services. This cause Cloud to be overloaded to serve a huge number of requests and therefore all its resources get exhausted, which can further origin DoS (Denial of Service). In DoS, in Cloud due to the presence of invader's overflow of requests, cloud becomes unavailable to serve legitimate user's requests. Nevertheless, strong authentication and authorization and IDS/IPS can offer defense in opposition to such type of attack.

Due to the overflow of requests or *zombie* attack, the Cloud provider has to provide more computational power so as to serve the huge number of requests (including *zombie* requests). By attacking merely on a one server, the attacker can cause an unavailability of a cloud service. This kind of an attack is called as DoS attack. It may perhaps affect other services of cloud too. Service instances on the cloud server are no longer able to carry out their projected tasks, if the cloud server's resources are totally worn out by processing the flood requests, which can lead into the whole Cloud system attaining a condition of full loss and is no more capable to serve any further service requests that are coming from legitimate users. This type of distributed attack is known as DDoS attack. So as to protect Cloud from these kinds of attacks one may can deploy a strong IDS/IPS system.

## **b) Injection attack**

Injection attacks are such attacks domain where intentionally malicious data is included as the input given to disrupt the normal functioning of the cloud. A few of the injection attacks are:

*XPath Injection:* When user and the cloud communicate, they communicate through XML files. XPath is a kind of query language for XML document as for relational databases we have SQL. Dissimilarity among SQL and XPath is that XPath is implementation independent [16]. XPath injection can occur by querying the XML database or when a service is invoked.

*SQL injection:* In such kind of attack user inject malicious SQL statements into an entry field for execution. SQL Injection is generally recognized as an attack vector for websites but can be used to attack any type of SQL database.

*Service Injection:* An opponent attempts to insert a suspicious service or new illegal virtual machine into the Cloud system and could supply malicious service to users. Cloud malware deform the Cloud services by changing (or blocking) Cloud functionalities. Let's take an example in which an opponent makes its own malicious services like SaaS, PaaS, or IaaS and attach it to the Cloud system. In case an opponent becomes successful to perform this, then legitimate requests are readdressed to the malicious services automatically. To protect against such an attack, there is a need to implement the service modules.

## **c) Man-in-the Middle attack**

An attacker is capable of accessing the data exchange between two parties, if SSL is not configured properly. In case of Cloud, an attacker is capable of accessing the data communication between data centers. To reduce or to prevent cloud from Man-in-the-Middle attack proper configuration of SSL and data communication tests between cloud and its client is required

## **d) Metadata spoofing attack**

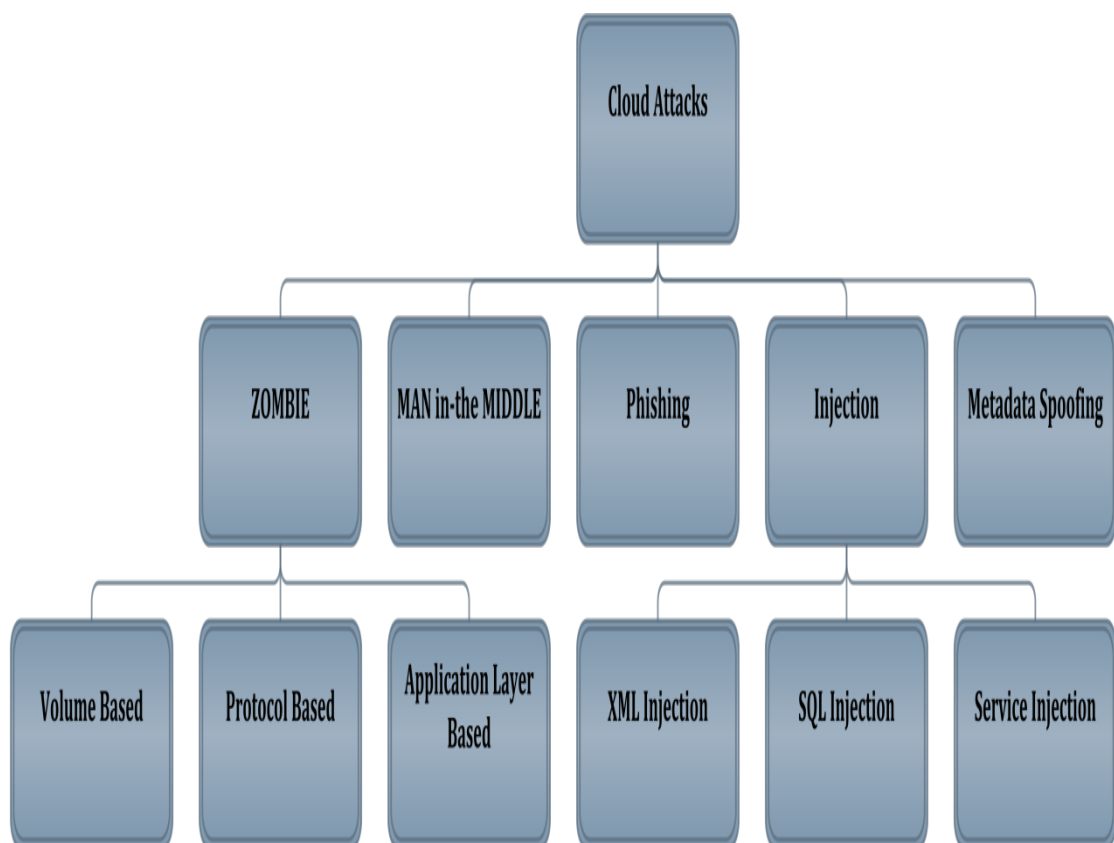
In such kind of attack, an opponent amends or changes the service's Web Services Description Language (WSDL) file where descriptions about service instances are stored. In case, if the opponent succeeds to suspend service invocation code from



WSDL file at delivering time, then metadata spoofing attack can be feasible. So as to triumph over such an attack, information about services and applications should be kept in ciphered form. Currently, [11] WS-Security Service is broadly used in cloud to endow with security for the system. In WS-Security, XML encryption and XML signature are used to provide data confidentiality and integrity. Well-built authentication (and authorization) should be imposed for accessing such critical information.

### e) Phishing attack

Phishing attacks are famous for manipulating a web link and sending a user to a false link to get confidential information. In Cloud, sometimes it might be possible that an adversary uses the cloud service to host a phishing attack site to hack accounts and services of other Cloud users.



**Figure4: Different Types of Attacks**

## 1.9 Security Services

When we access the cloud, there are some major security issues that needed to be resolved. Those are:

### a) Data integrity

It is the assurance that the data is received is as same as sent by an authorized entity which implies there are no modifications, no insertions, no deletions are there in the data.

Attackers try to tamper with the information and change its content which poses major security threat to the data and its user.

### b) Data confidentiality

It is the protection of data from the unauthorized discloser. Attacks keep an eye on the data or the information being transferred over the network. Sometimes they don't tamper the information; they just extract the crucial information and misuse that information (example- extracting the user id and password-Table1). So data confidentiality is a major security concern in SaaS cloud.

### c) Authentication and Authorization

Authentication is the assurance that the communicating entity is the one that it claims to be that is "who are you?" and authorization is a kind of access control "what you can do." This means that no unknown or harmful entity should be able to pretend that he or she is the authenticated one and even authenticated persons should have a limited access to the data.

**Table1: Attacks with Example**

Attack	Affected Security Mechanism	Example
<b>Spoofing</b>	Confidentiality	Illegitimately using someone's credentials
<b>Tampering</b>	Integrity	Illegally changing, modifying or altering the data
<b>Repudiation</b>	Audibility	Performing illegitimate operation in a system that lacks ability to trace it
<b>Denial of Service</b>	Availability	An adversary gains control of a tenant's VM, and makes another's Web server unavailable.

## **1.10 Problem Statement**

- To address different security concerns.
- To propose an efficient encryption algorithm for the cloud computing environment.
- To enforce the confidentiality on messages as well as on data.

## **1.11 Thesis Organization**

The organization of thesis is as follows in Chapter 1: Introduction, we are going to discuss the basic about cloud computing, its characteristics, different types of threat and attacks on cloud. The next chapter that is contained in this thesis is Chapter 2: Literature survey. In this chapter we have tried to cover as much as papers on types of attacks on cloud, techniques to embed security and different encryption schemes. The further comes the Chapter 3: Proposed Work, in this particular chapter we have discussed our proposed XML DNA encryption/decryption approach and also discussed about various other old encryption schemes. Then next comes Chapter 4: Implementation and Results, in this chapter we have discussed the implementation details and have shown various results that are coming from our proposed work. At last comes the Chapter 5: Conclusion and Future Scope.

# Chapter 2

## LITERATURE SURVEY

*Cloud Security*

*XML and Attacks*

*Embed Confidentiality*

*Encryption Techniques*

## **CHAPTER - 2**

### **LITERATURE SURVEY**

In this section we have discussed the literature review regarding security of XML files, attacks both on cloud and on XML files, different methods to implement confidentiality, some old and new encryption techniques.

#### **2.1 Cloud Security**

Hamlen et.al in [35] discuss security issues for cloud computing and present a layered framework to secure clouds and then focus on two of the layers, i.e., the storage layer and the data layer. The issues include storage security, middleware security, data security, network security and application security. The main goal is to store the data securely and manage the data not under control by owner of the data. A bottom up approach to security is proposed where work is done on small problems in the cloud that we hope will solve the larger problem of cloud security. Firstly, how documents can be secured is discussed so that they may be published in a third party environment. Next thing was that how security can be enhanced by using secure co-processors. It is found that due to complexity of cloud it is difficult to achieve end-to-end security. Even if some parts of the cloud fail the challenge is to ensure more secure operations. Building trust applications from untrusted components will be a major aspect with respect to cloud security. Hu et.al in [31] present a survey on the architectures, concepts and challenges of cloud computing. A summary of challenges in cloud computing with respect to security, virtualization, and cost efficiency are discussed. Among the stated issues, security issues is the most important challenge in the cloud computing. Chen and Zhao in [37] discussed about the analysis on data security and privacy protection issues associated with cloud computing across all stages of data life cycle along with some solutions. One concern is that what information to reveal and who can access that information over the Internet. Another concern is whether web sites which are visited collect, store, and possibly share personal information about users. Privacy can be achieved by separating sensitive data from non-sensitive data followed by the encryption of sensitive elements.

Current security solutions for data security and privacy protection are discussed below.

- a) Roy I and Ramadan developed privacy protection system called airavat that can prevent privacy leakage without authorization in Map-Reduce computing process.
- b) A fully homomorphic encryption scheme was developed by IBM in June 2009. It allows data to be processed without being decrypted.
- c) A key problem for data encryption solutions is key management. On the one hand, the users have not enough expertise to manage their keys. On the other hand, the cloud service providers need to maintain a large number of user keys. The Organization for the Advancement of Structured Information Standards (OASIS) Key Management Interoperability Protocol (KMIP) is trying to solve such issues.
- d) NEC Labs's provable data integrity (PDI) solution can be used for public data integrity verification.
- e) Mowbray proposed a client-based privacy management tool for data storage and use stages. It provides a user centric trust model to help users to control the storage and use of their sensitive information in the cloud. A privacy protection framework was proposed by RandikeGajanayake based on information accountability (IA) components. The identification of users who are accessing information and the types of information they use is done by IA agent. In case of misuse being detected, the agent defines a set of methods to hold the users accountable for misuse.

Tianfield in [36] discusses about the various issues of security in cloud computing. In the paper cloud security requirements are analyzed in terms of fundamental issues like trust, availability, audit, integrity and confidentiality. As security is a major issue, it should be applied at different levels to ensure right implementation of cloud computing such as: security of host server, security of data storage, network security and security of application. Cloud security can be analyzed along three features: *Identity security* - Key elements of cloud security is identity management at end-to-end level, authentication from third party and federated identity. The integrity and confidentiality of data and applications is preserved by identity security while

providing access to appropriate users. *Information security* - Information security is closely related to third-party data control. Concerns regarding information security include the way in which data is stored and accessed, compliance and audit requirements. All sensitive data including archive data, needs to be segregated properly on the cloud storage infrastructure. Encrypting and managing encryption keys of data in transit to the cloud or data at rest in the service provider's datacenter is critical to protecting data privacy and complying with legal and regulatory mandates. 3) *Infrastructure security* - The foundational infrastructure for a cloud must be inherently secure whether it is a private or public cloud or whether the service is SaaS, PaaS or IaaS. The cloud computing infrastructure, including servers, switches, routers, storage devices, power supplies, and other components that support operations and transaction of data and information, should be physically secure. Unauthorized user or employee should not be able to access any component.

## **2.2 XML and Attacks**

Chau discussed many security threats and security attacks [17]. The author covered into two broad terms that are malware injection attack and wrapping attack. These malware injection attacks include cross-site scripting attacks and SQL injection attacks. Wrapping attack covers the attack on SOAP messages that break confidentiality between client and server. To insure confidentiality among clients and servers WS- Security for web service is applied and some other counter measures are also mentioned by the author. Meiko et al discussed many technical issues. If we use XML signature for the authentically purpose then it is prone to XML attack known as XML Signature Element Wrapping attack. Author has also discussed some of the Browser Security issues that are cloud are not protected in current browser based authentication protocols. One of the reason is browser that browser by itself is unable to issue XML based security tokens as a solution author told to add XML encryption and signature as an enhancement to browser security API [23]. Youxiang and Yang [24] have classified SQL injection, XPATH inject and XSS as confidential vulnerabilities. In this paper author discusses that XPath injection attack aims at XML document. Attacker can gain whole information through XPath query on XML document even without any aprior knowledge about XPATH query. By operating XPath queries an attacker can control XML database. This attack can expose all the

confidential information. Subashini and Kavita [16] discussed that SaaS applications are multi-tenant in nature and moreover they are hosted by third party. While using Web services, application exposes their functionality. There are many challenges left that are still needed to be resolved, out of which one of the biggest challenges that are still needed to be tackled within web services is managing the transaction. In market there are lots of standards available for example WS-Transaction and WS-Reliability but they are not mature yet.

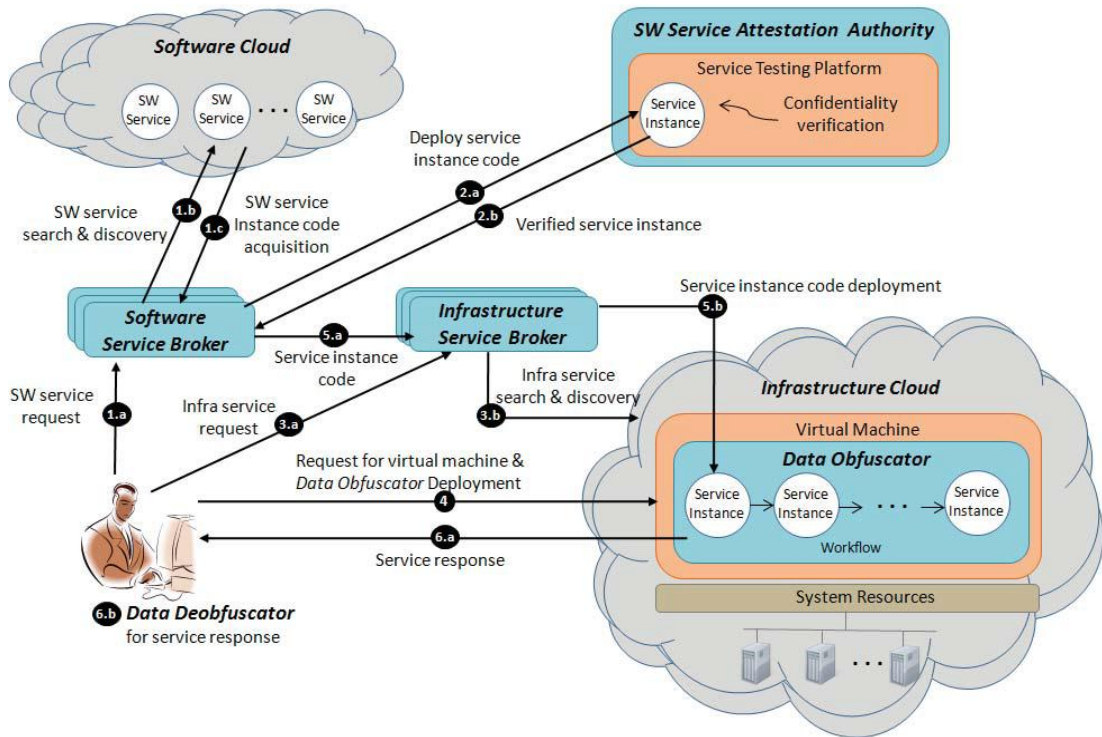
As the clients are communicating to the cloud via internet then all communication is done through SOAP messages. So these SOAP messages are required to keep confidential which are in XML format. There are different techniques to keep those messages confidential. Some of these techniques are listed in Yue-Shenet al. [12]. Author describes core web security technologies as XML signature and encryption. Integrity in the document can be applied through the implementation of XML Signature and which can also realize the identification authentication, but there is the great requirement of the confidentiality. SSL carries encryption to the complete information, but don't leave the alternative to encrypt to information partially, which is a severe performance issue when we are transmitting the huge amount of information. Moreover SSL only provides point-to-point security, but is not capable to provide the end-to-end security.

### **2.3 Embedding Confidentiality**

Bhosale et al. [34] provides a 3 dimensional framework along with digital signature and RSA algorithm where the user will upload the data over cloud based on the various security levels. Protection ring 1 will provide high level of security, ring 2 will provide less security and ring 3 will provide least level of security. Security of cloud is enhanced by using this framework with RSA and DSA algorithm combination. Availability of data is achieved by overcoming many existing problem like denial of services, data leakage. It also provides more flexibility and capability to meet the new demand of today's complex and diverse network. Digital signature is a scheme that checks the authenticity of the document or a message. If a message is created by known user, then the digital signature will send a receipt to the sender stating that the message was not altered. An asymmetric type of cryptography is employed by the digital signature. When a digital signature is properly implemented

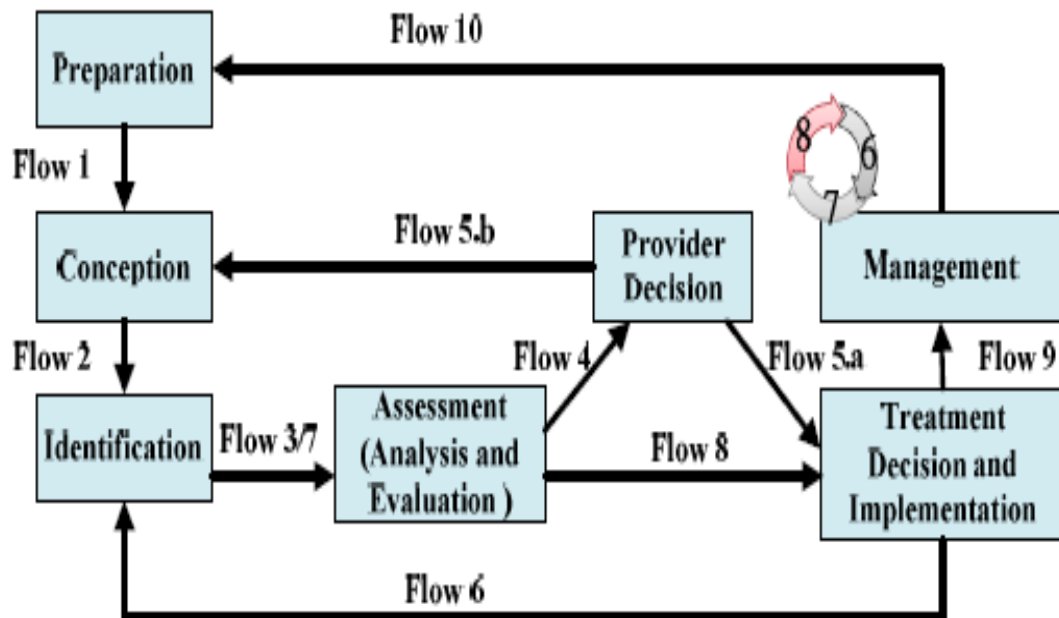


then even if messages are sent through a non secure channel, the digital signature gives the receiver reason to believe that the message was sent by the claimed sender. Non-repudiation is provided by the digital signature, meaning that the signer cannot claim they did not sign a message. Even while claiming their private key remains secret. Some non-repudiation schemes offer a time stamp for the digital signature, so that even if the private key is exposed, the signature is valid nonetheless. Digitally signed messages may be anything that can be represented as a bit string. Examples: electronic mail, contracts, or a message sent via some other cryptographic protocol. RSA supports encryption and digital signatures. RSA gets its security by integer factorization problem. It is easier to understand and implement the RSA algorithm. There are many techniques [4] for protecting users' data from outside attackers, but currently no effective way is available for protecting users' sensitive data from service providers in cloud computing environment. **Techniques for confidentiality** protection are access control, encryption, identity management, use of strong keys, integrity assurance etc. however these mentioned techniques cannot provide confidentiality protection in the mentioned situation because they were developed only for protection from malicious third party outside the systems. Since the cloud computing systems have service providers inside the system as a new thread to cloud computing. So, in 2010 Stephen S. Yau et al. the authors presented an approach to secure the privacy of clients' data and information from cloud service providers and make sure that service providers can't accumulate clients' private data while the data is processed and stored in cloud computing systems. Their approach has three major aspects. (1) Separating software service providers and infrastructure service providers in cloud computing. (2) Hiding information about the holder of the data, and (3) **Data obfuscation** as shown in Figure 5. Here, data obfuscation is a method used by an infrastructure cloud in which the user data is processed such that users confidential information is not revealed to its Infrastructure service provider. There are a variety of other frameworks that ensure confidentiality of the data from the attacks and also from the various service providers. As there is another framework presented by the authors in 2011 by Chou et al. in which paper [8] proposes a complete supply approach, SaaS Confidentiality Risk Management (**SCoRiM**) Framework as shown in Figure 6 which ensures protection of critical data in small and average sized companies and helps to decide whether which kind of public SaaS provider best suits their needs of confidentiality.



**Figure5:Data Obfuscation**

Its goal is to increase the data confidentiality management along with and without support from the providers all through the whole SaaS integrated system development life cycle.



**Figure6:SCoRiM Framework**

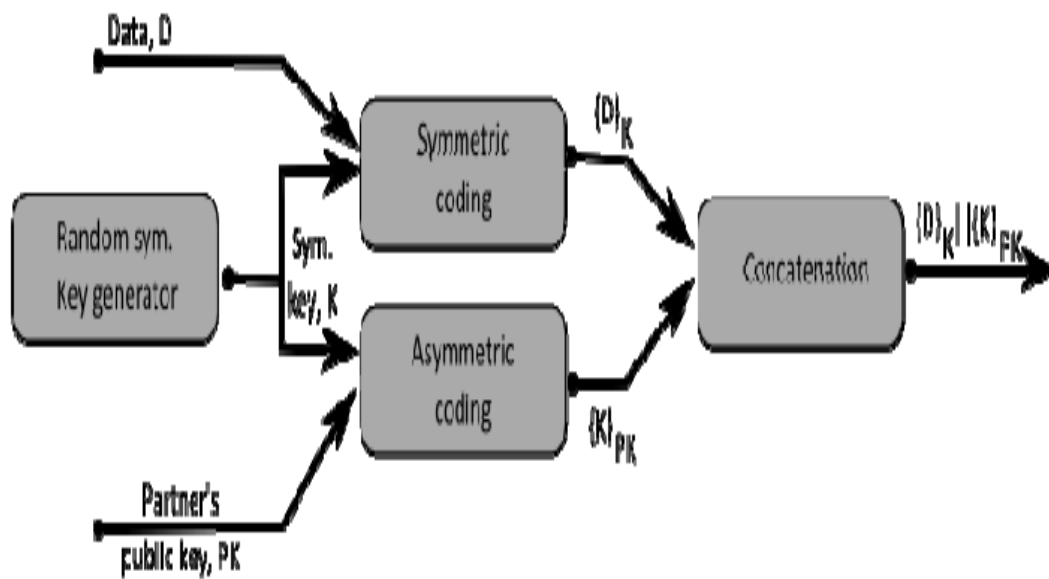
## 2.4 Encryption Techniques

For XML encryption we can employ any new or old encryption algorithms. As discussed by Thakur and Gupta [7], they brought the idea of dividing the data to multiple clouds according to the level of integrity and confidentiality required to them and then apply encryption techniques. Traditional encryption techniques like Advanced Encryption Standards (AES), Data Encryption Standards (DES), and Digital Signature Standards (DSA). AES, DES is applied where less integrity is required like downloading, surfing websites and RSA is applied where high integrity is required like in banking transactions, ATM transaction. Also the author has made comparisons between four of these algorithms and concluded that first AES is better than DES because key length is optional and larger than AES. Second RSA is better than DSA reason behind is that DSA only provides authentication but RSA provides both authentication and encryption. In this paper [26], author captures some major differences between the traditional cryptography and DNA cryptography Technique. The paper discusses some important aspects of both the cryptographic techniques and pointed out some of the major advantages & disadvantages of both the techniques as shown in Table-2.

**Table 2: Comparison between Traditional and DNA cryptography**

Cryptography	Security	Time Complexity	Storage Medium	Storage Capacity	Stability
<b>Traditional</b>	1 Fold	$\geq$ Seconds	few (Silicon Strands)	Computer (Silicon chip carries 16MB)	Dependent on implementation environment
<b>DNA Cryptography</b>	2 Fold	$\geq$ Few Hours	DNA strands	1 gram of DNA strand carries $10^8$ TB	Dependent on environment conditions.

No doubt that DNA has such a bright future in the field of cryptography. But as discussed by Pruthi and Dixit there are also some drawbacks of DNA cryptography [27] and that are High computational complexities, huge computing time and it also requires high technical bimolecular laboratories. Hossein in his work discusses that data is encrypted via DNA sequences using modified Haffman technique using selective encryption method [25]. Selective encryption is a method where a part of message is encrypted keeping the remaining part unencrypted, can be a viable proposition for running encryption system in resource constraint. Encrypted data is stored in look up table or in compressed file by using ASCII code and online library file acting as a signature. The running time of this algorithm is tested on benchmark DNA sequence. The running time of this algorithm is very few second and the complexity is  $O(n^2)$ . In the literature some new cryptographic techniques are also present that ensure confidentiality. Liu and Lin [18] have discussed a new cryptographic technique i.e. DNA encryption technique and embed in word document to assure confidentiality. First of all the plain text is encoded by a DNA sequence. The second step is the equal length DNA sequence created by Chebyshev maps is used to encrypt the already considered DNA reference sequence then he attached the result to the elementary DNA sequence. Then next step is to shift the whole DNA sequence for infinite times, then author insert them into the word file by amending the fore color of the characters. Every whole data contained in the word file can be entrenched in one character that is a 6-bit DNA cipher. The chebyshev maps and shifting times are all considered as plaintext and keys which can be successfully pull out from the host document. Terec et al. [11] have discussed the implementation of various cryptographic techniques in Java, Maltab and BioJava and also explains how DNA encryption is implemented in three of them. Based on confidentiality properties these algorithms are used. Author also compares symmetric and asymmetric DNA encryption techniques and made comparisons among them on various platforms (vista, windows7 etc.).He also captures differences like Asymmetric DNA (Figure7) is more reliable than Symmetric OTP DNA. DNA requires longer time than other encryption techniques like AES, DES, 3DES.



**Figure7: Asymmetric DNA Scheme**

In this paper, authors also discussed an asymmetric DNA scheme in which they collaborate three encryption technologies. This algorithm starts when both the initiator and the other party generate a pair of asymmetric keys. In the next step, both about confer about the symmetric algorithm and its specification and also the sequence to use in, where the indexes of DNA bases will be looked upon. Ranalkar and Phulpagar discussed that user's crucial data is fragmented into parts and is distributed over multiple cloud service provider (CSPs) such as to attain better data availability and security [5] and then apply DNA sequencing algorithm on it. In their research they found that we need to distribute it to many clouds as possible because as the CSPs are increased to 12 the execution time is reduced 4 times and as small as the data is its difficult to capture the crucial information from it. So as the CSPs increase execution time decreases and security increases.

# Chapter 3

## **PROPOSED WORK**

*Cloud Confidentiality*

*XML Encryption*

*DNA Encryption/Decryption*

*XPath Attack*

## **CHAPTER-3**

### **PROPOSED WORK**

Cloud computing model provides us with three service modules – Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). These services act as an end point of connection. The communication between the customer and these services takes place through XML files. If we take the example of filling of a form then the entries filed by the client are transferred over the network through these XML files. Also the communication among Web Service and the clients is mainly done through plain-text XML formats like SOAP messages and WSDL [13]. Moreover there are some companies that store their data as XML database due to its document-centric data structure [20] and it avoids the overhead of disbanding XML only for storage purpose, and of reconstructing the XML when applications need to read data. So all the users interact with the cloud using internet via these XML files then these files and the content of these files need to be protected so as to safely transfer the confidential information. There are different mechanism by which we can secure XML messages and the mechanisms are SSL, XML Signature and XML Encryption.

#### **3.1 XML Encryption**

XML Encryption which is also well-known as XML-Enc., is basically managed by the W3C recommendations that define how we can encrypt the contents of an XML. XML Encryption grants integrity and confidentiality to the messages that are being transmitted between the cloud user and the cloud provider. These messages can be of any type like request/response messages, notifications etc. The cryptographic technique we have used in our work is DNA encryption and decryption.

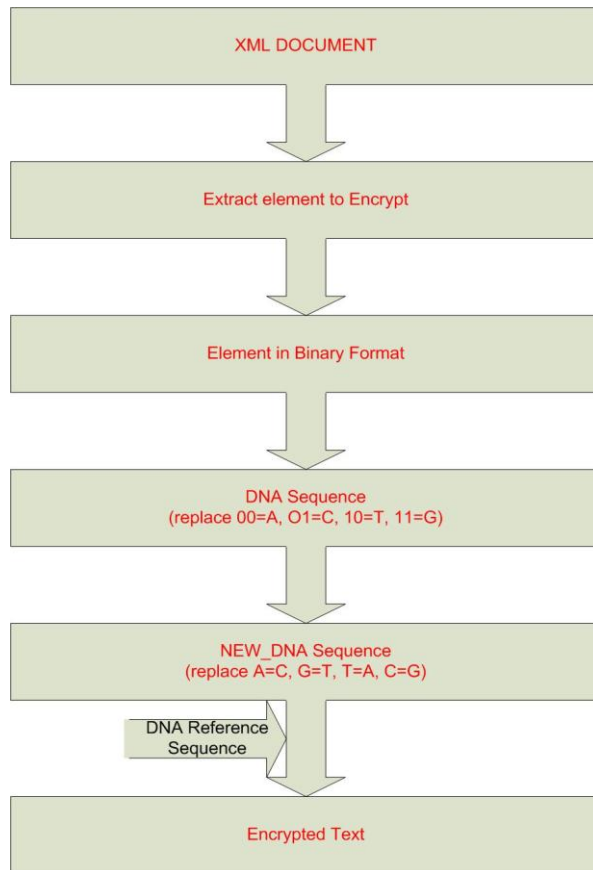
Besides XML encryption we have SSL and XML signature to secure the internet transmission. But to embed confidentiality XML encryption is best because SSL can vary encryption to the complete information, but doesn't have the choice to realize the encryption to the partial information i.e. there is no selective encryption in SSL. When

transmission of mass data is taking place then it will cause the serious performance question. Moreover the SSL only guarantees point to point security, but is not capable to safeguards the end-to-end security. We have another technique for embedding security is XML Signature. XML signature may possess the integrity of the data and also may carry the identification authentication. But the data also needs the confidentiality which is provided by XML Encryption. When the client wishes to interact with the cloud, the communication requires confidentiality because the communication may contain some confidential information say password or credit card details etc. So communication needs to get secured. That is the reason we need to encrypt the messages. The algorithm used by us to embed confidentiality here is XML DNA Encryption/Decryption. This DNA encryption algorithm will selectively encrypt the required information that is needed to keep confidential and hence provides the confidentiality in the cloud environment. By this XML DNA encryption we can secure the communication among cloud and client. It will protect the communication from improper information disclosure, information leakage and from various attacks over the network.

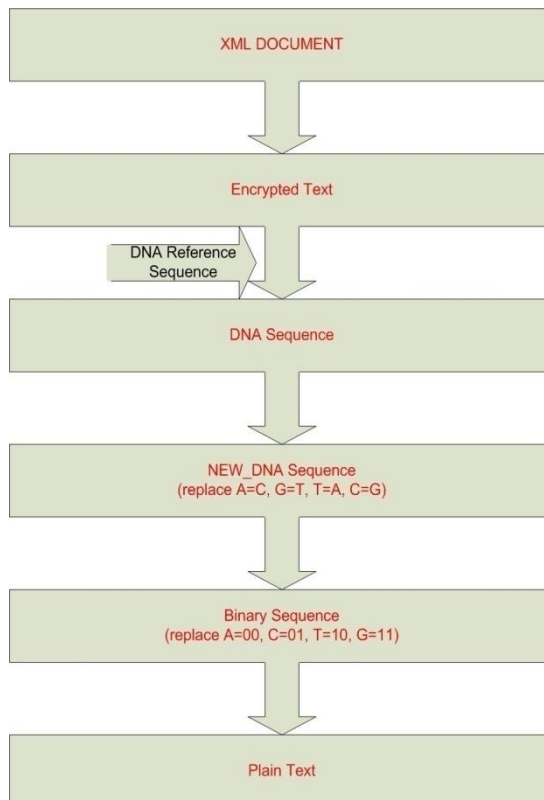
### **3.2 DNA Encryption/Decryption**

In this algorithm, firstly we extract the required element from XML file. After that the further step is to convert the extracted element in binary form. Then we assign binary combination to the DNA bases (A=00, T=01, C=10, G=11) to the extracted text. Now what we have is a DNA sequence consisting of DNA bases that are A, T, C, and G. Further is to apply complementary pairing rule on this DNA sequence that is to replace the bases with their complements (A=C, C=G, G=T, T=A). The next step is to take a DNA reference sequence from a gene sequence database. A one complete DNA sequence consists of 20 pairs of bases. Now we have two DNA sequences one is encoded DNA sequence and the other one is actual DNA sequence from genes database. So the last step is replace the encoded DNA sequence, base pair wise, with the number of occurrence of the corresponding base pair in the actual DNA sequence. The algorithm steps are as follows: (also shown in Figure8 and Figure 9). The whole step by step sequence of algorithm with sample output is shown in Table-3





**Figure8: DNA Encryption**



**Figure9: DNA Decryption**

### 3.3 DNA Algorithm

#### DNA Encryption

**Input=Selective Plain Text from XML file, DNA Reference Sequence**

**Output-Encrypted Text in XML file**

Step1: Replace Plain Text (PT) by its binary

BinaryText←PlainText,

Step2: Assign binary to DNA bases (A=00, C=01, T=10, G=11)

Step3: Replace binary digits of both the text and key by its DNA bases

DNASeq1 ← BinaryText

Step5: Replace DNA sequence with complementary bases (A=C, G=T, T=A, C=G)

NewDNASeq← DNASeq1

Step6: Take the input reference sequence and replace DNASeq by the base pair occurrence number

NumericalEncrypted text ← NewDNASeq

Step7: Replace it in same XML file

#### DNA Decryption

**Input- EncryptedXML file, DNA reference Sequence**

**Output: plain text in XML file**

Step1: Take encrypted text from XML file

Step2: Replace encrypted text by DNA bases according to the reference sequence

DNASeq1← Encrypted text

Step3: Replace DNASeq by its Complements

DNASeq2← DNASeq1

Step4: Assign binary to DNA bases (A=00, C=01, T=10, G=11)

Step5: Convert DNA text to binary

BinaryText← DNASeq2

Step6: Convert Binary to its actual text

Plain Text← BinaryText

Step7: Replace the plain text in XML document

This encryption technique is very effective rather than other old encryption techniques as now the attackers are well aware of the old techniques. In our case the probability to judge the correct plain text is .0000000061 because there are approximately 163 million of DNA sequences accessible freely [21].

**Table3: DNA Sample Output**

Input	Output
<b>XML- File</b>	1234 (selected element )
	00110001001100100011001100110100 (binary conversion)
	AGATAGACAGAGAGTA (G=11, C=10, T=01, A=00)
	CTCACTCGCTCTCTAC (A=C, C=G, G=T, T=A).
<b>DNA Reference: [TA<sub>1</sub>], [GC<sub>2</sub>], [TG<sub>3</sub>], [AG<sub>4</sub>], [CT<sub>5</sub>], [CT<sub>6</sub>], [TT<sub>7</sub>], [TG<sub>8</sub>], [AC<sub>9</sub>], [TC<sub>10</sub>], [TC<sub>11</sub>], [TA<sub>12</sub>], [AT<sub>13</sub>], [CA<sub>14</sub>], [CC<sub>15</sub>], [CC<sub>16</sub>], [TC<sub>17</sub>], [CG<sub>18</sub>], [TG<sub>19</sub>], [CT<sub>20</sub>]</b>	201420182020209 (encrypted text)

### 3.4 Comparison

To check the efficiency of the proposed work, we have compared our algorithm with some existed encryption techniques. In literature, various symmetric and asymmetric encryption techniques are present for instance AES, DES, RC4, 3DES are some of the famous symmetric encryption techniques and RSA ECC and many more are asymmetric algorithms. In the presented work the comparison of proposed work has been done with Data Encryption Standard (DES), Advanced Encryption Standard (AES) and Rivest Shamir Adleman(RSA).

#### a) DES

Data encryption standard (DES) is an exemplar of Symmetric cipher. So it's working is based on to use the same key while encrypting and decrypting the text which means sender and receiver must utilize the same private key. It is a block cipher which means both key and DES are performed to a block of data simultaneously instead of one bit at a time.

The encryption begins by partitioning the plaintext into blocks of 64-bit [29]. Every block is encrypted using the private key via performing operations of permutation and substitution on it (Figure 10). This whole procedure includes 16 rounds and can also run in four different modes, individually encryption blocks and making every cipher block dependent on all the previous blocks. The decryption procedure is just the inverse of encryption and the order in which the key is applied is also reversed.

## b) AES

Advance data encryption is an example of symmetric cipher. For the process of encryption and decryption both the sender and the recipient use a single key. In this the length of data bock is preset as 128 bits but key length may vary as 128,192, or 256 bits respectively [28]. It is an iterative algorithm. Here each iteration is known as round and the net number of rounds are  $N_r$  and that can be  $N_r=10, 12,$  or  $14,$  according to the key length 128,192, or 256 bits. The 128- bit data block is subdivided into the blocks 16 bytes each. Then mapping of bytes takes place to a  $4*4$  array known as the state. In this encryption scheme all rounds consist of four transformations those are SubBytes, ShiftRows and MixColumns and the RoundKey but the last round excludes MixColumns operation(Figure11).

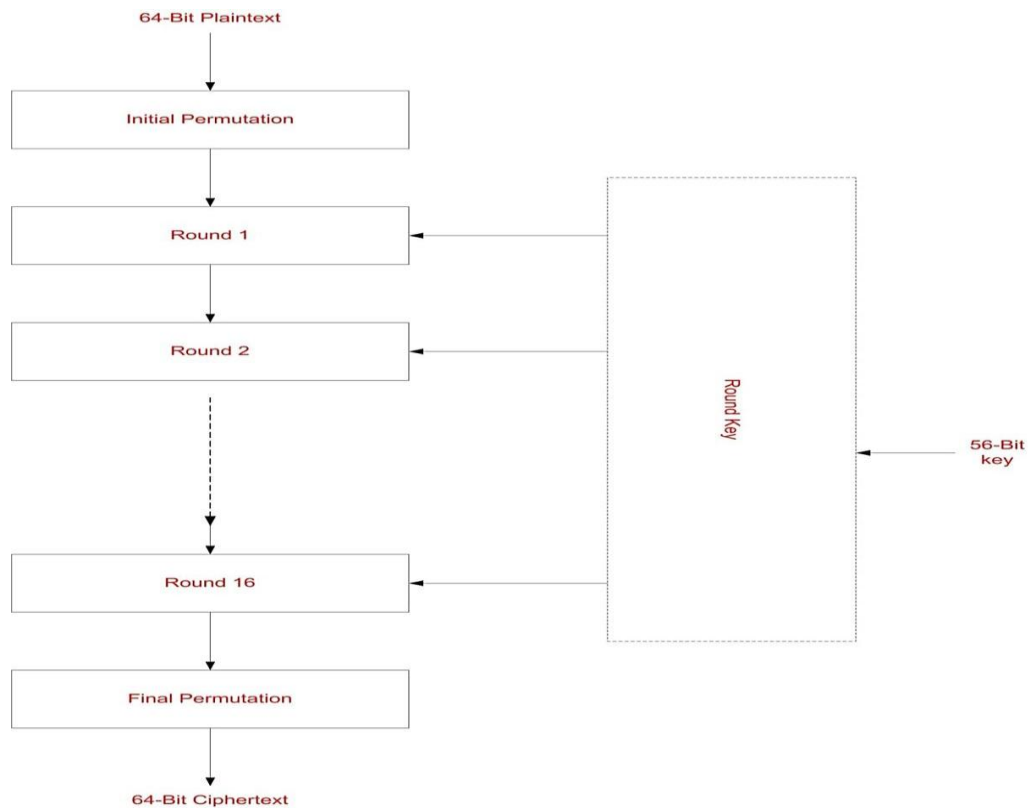
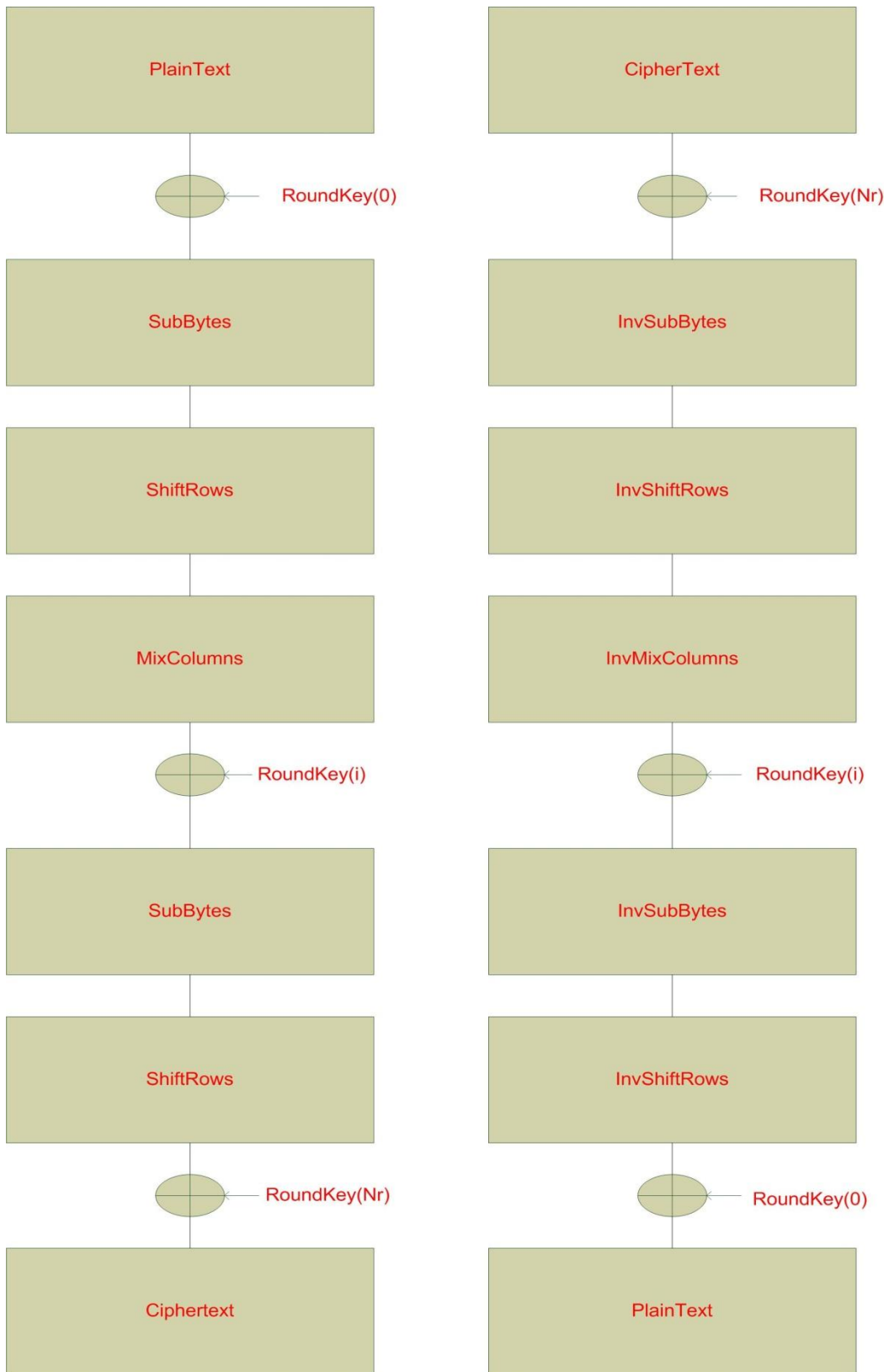


Figure10: DES Encryption



**Figure11 : AES Encryption, Decryption**

### c) RSA

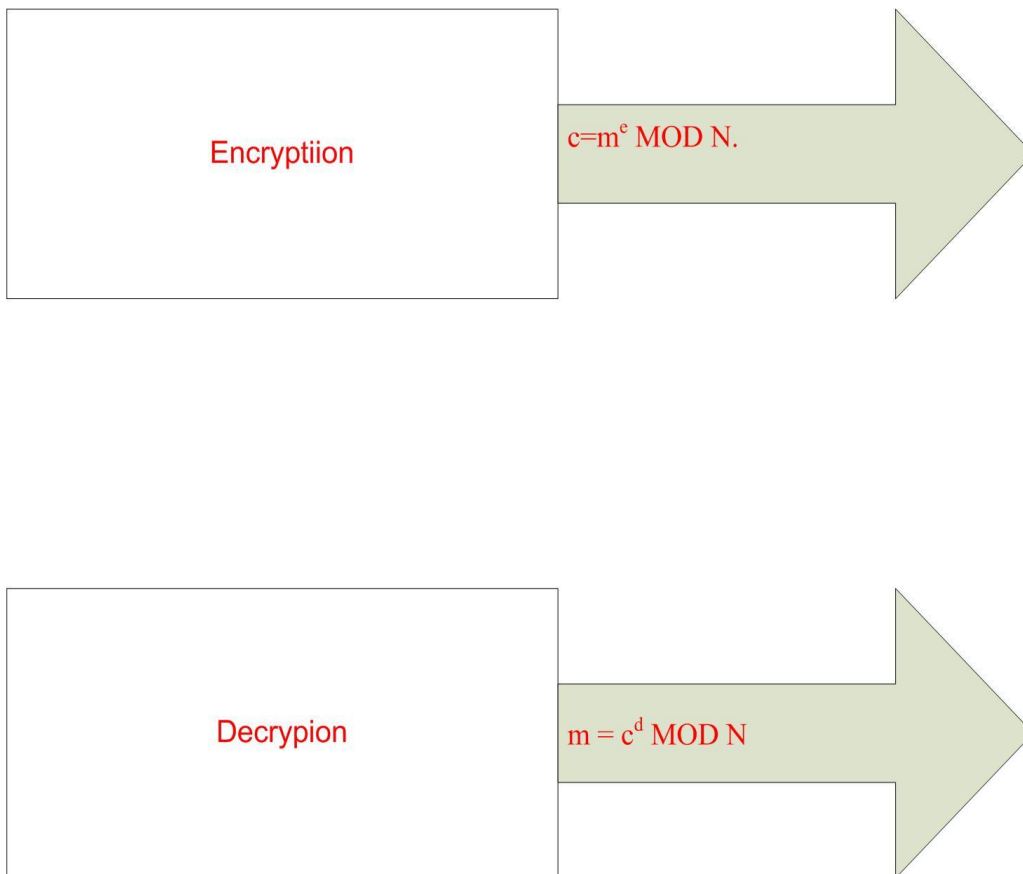
Rivest, Shamir, and Adleman algorithm,(Figure12) and is famous as RSA algorithm which is a asymmetric, public key algorithm. In this algorithm user's confidential information consists of two prime numbers  $p$  and  $q$ . User computes its key through the product of  $p$  and  $q$ ( $N=pq$ ) and a number  $e$  such that this  $e > 1$  which is coprime to  $p-1$  and  $q-1$ . So as to transmit the message sender partition the message into blocks  $c_i$ (numbers in the interval  $[1, N-1]$ ) [30]. Finally in order to encrypt a block  $c$ , the sender utilizes the public numbers  $N$  and  $e$  so as to create)

$$c = m^e \text{ MOD } N.$$

$N=pq$  factors are already known to the receiver and through this he is capable of calculating the decipher key which is

$$e \cdot d = 1 \text{ MOD } (p-1)(q-1)$$

and at last the receiver decipher it using  $m = c^d \text{ MOD } N$



**Figure12: RSA Encryption,Decryption**

### **3.5XPath Injection Attack**

As the user input his information on his end, and then this information is transferred to the cloud over the network via internet using XML files. These files may contain extremely confidential information that need to be concealed from various threats, thefts and attacks. Attackers may try to fetch out the information and can harm the customer in many different ways. There are many possible XML attacks e.g. man-in-the-middle attack, SQL Injection attack, XPath Injection, XDoS and many more other attacks [22]. In this work we will be focusing on XPath Injection attack. XPath Injection attacks take place when a web site uses user-supplied input to form an XPath query for an XML data or a company stories its data in an XML format. By inserting deliberately deformed information into the web site, an attacker can discover the structure of XML data, or can expand access to other privileges of the website. He then may be capable to get his privileges of the web site if the XML data is being worn out for authentication purposes. Moreover the attack can also take place on an XML database. If the attacker knows the fields of XML database, the attacker can even fetch the complete XML database. As we know an XML files has no privilege system or ant mechanism of access control within it. By performing XPath injection attack by firing XPath queries, there are possibilities for attackers to dig out the whole XML database [22]. By doing such an act, an attacker not only breaks the privacy of a user but also breaks the confidentiality of not only the company but of also the user of that company too. This could be very harmful to the client as there can be information disclosure of confidential information of the user. An attacker may use the confidential information to harm client in numerous ways e.g. by knowing username and password getting access to a user account, by knowing account pin of ATM he can do many illegal transactions and there are many other harmful acts that can be executed by an attacker.

Attackers perform XPath attack by sending malicious input into the XPath query. XPath is a syntax which is used to illustrate component of an XML document. By XPath, we can refer to any element, attribute of the elements, all specific elements that contain some text and many other variations. Attackers take advantage of bad code sectors, unsanitized input, no proper input validations, no encryption on data fields of XML to perform the XPath attacks. Attackers exploit the XPath queries in

many ways, for instance let us take the example of a user login or register system on a web page. If no proper validations are applied on inputs, the attacker may exploit by feeding random inputs or smart inputs that may work in his favor.

The one solution proposed by us is to strongly encrypt the confidential input so as on fetching the inputs through xpath queries, the attacker get encrypted inputs. Now it depends on the encryption algorithm chosen by the programmer that how strongly it encrypts the data, how the programmer is implementing it. Let's take the example of database file that consists of fields like first name, last name, card pin of a user. Now let's demonstrate the attack and its affect before and after encryption. Let's say that we have an XML file that contains first name as tom, last name cat and credit\_card as 1234. Now an attacker wants to fetch the details of a person say firstname, lastname and credit pin then its XPath query and its result before encryption is explained below

#### **XPath query:**

```
XPathExpressionexpr = xpath.compile("//employee[firstname/text()=' ' and lastname/text()=' ']/credit_card/text()")
```

```
Feeding first name = tom
```

```
    Last name = cat
```

By inserting the first and last element an attacker may able to fetch the card pin if and only if that first name and last name matches the xml database. By feeding the input into the query the XPath query becomes

```
XPathExpressionexpr =  
xpath.compile("//employee[firstname/text()='tom' and lastname/text()='  
cat']/credit_card/text()")
```

Next step is to evaluate the query

```
result = expr.evaluate(doc, XPathConstants.NODESET);
```

If the firstname as “tom” and last name as “cat” is present in the XML file then the query will return credit\_carddetails. As per in our case the query will result in as 1234, as there is no encryption applied on element credit\_card. Now again running the same query but now after the DNA encryption was done on XML file:

Query:



```

XPathExpressionexpr =
xpath.compile("//employee[firstname/text()='tom' and lastname/text()='
cat']/credit_card/text()")

```

Evaluate:

```
result = expr.evaluate(doc, XPathConstants.NODESET);
```

**Result: 201420182020209**

**Table 4 : Attack Before Encyption:**

Query	Result
<pre> exprxpath.compile("//employee[firstname/text()='tom' and lastname/text()='cat']/credit_card/text()"); </pre>	1234
<pre> expr = xpath.compile("//employee[firstname/text()=' '1'='1' and lastname/text()=' '1'='1']/credit_card/text()"); </pre>	1234
<pre> or '1'='1' and lastname/text()=' or '1'='1']/credit_card/text()"); </pre>	1111

**Table5 : Attack After Encryption**

Query	Result
<pre> exprxpath.compile("//employee[firstname/text() ='tom' and lastname/text()='cat']/credit_card/text()"); </pre>	201420182020209
<pre> expr = xpath.compile("//employee[firstname/text()=' 2014201420142014 or '1'='1' and lastname/text()=' or '1'='1']/credit_card/text()"); </pre>	201420182020209

Here in Table 3 and Table 4 it is clear that how encryption is preventing information leakage and maintains confidentiality of cloud user. This result is a DNA cipher. Now

if the attacker wishes to know about the credit card pin of the person then he needs to decrypt it first. As mentioned earlier the probability to decrypt it with a correct guess is .0000000061 and this can be further reduced because we have four DNA bases A,C,T and G , now assigning them binary digits as A=00, C=01, T=10 and G=11 can take 4! combinations. Moreover in the complementary rule we have A=C, G=T, T=A, C=G, this can also take 4! combinations. This will result as .0000000061/(4!\*4!), which gives us the probability as  $1.065098841 \times 10^{-11}$ .

# Chapter 4

## Implementation

*CloudSim*

*Analysis of Proposed Approach*

*Comparisons*

## CHAPTER-4

### IMPLEMENTATION and RESULTS

An XML encryption has been proposed in order to improve the confidentiality of client and the cloud. Different encryptions schemes have been used to achieve the results. An algorithm has been proposed to implement the cryptographic algorithm for encryption. The analysis of the proposed and the traditional cryptographic algorithm has been done using NetBeans IDE 7.4 on 32 bit windows7 environment with 4GB RAM. To estimate and plot the output analysis is done using Matlab with the same system configuration.

In order to provide assurance of some crucial characteristics in cloud systems like reliability, security, fault-tolerance, sustainability, and scalability computational services well-timed, repeatable, and convenient methodologies are required for assessment of new cloud policies and applications before the actual development of cloud products [33]. **Simulation** is a supple methodology that is used for analysis of performance of a present or proposed any kind of company's activity, new product or application, manufacturing line etc. **Performing simulations** and **analyzing the results**, helps to know the functioning of the present system, and what would happen if changes are made to it – or estimation of behavior of the proposed new system is done. IT companies have various benefits of using simulation based approaches by allowing:

- Testing of their applications in a repeatable and controllable environment.
- Before deploying on real clouds tuning of system bottleneck should be done
- For creating, testing and deploying adaptive services techniques experimentation with various workload mix

In order to evaluate the proposed algorithm to embed security simulation is done via utilizing a tool named “CloudSim”.

## 4.1 Introduction to CloudSim Simulation

CloudSim is an emerging framework that facilitates simulation, modeling, and experimentation of cloud computing infrastructures and management services. Silent features of CloudSim incorporates [32]

- To model and simulate large scale data centers in cloud environment
- To model energy aware computational resources and simulate it
- Data center network topologies and message passing applications are modeled and simulated.
- Simulation elements inserted dynamically, simulation stops and resumes feature.
- The environment is very user friendly in terms of user can specify policies for allocation of hosts to Virtual Machines and policies for allocation of host resources to virtual machines

Some of the major benefits of CloudSim are:

- Time effectiveness
- Flexibility and applicability
- Test policies in repeatable, controllable, and manageable environment
- Adjust system limitations before deploying on real clouds

In our work first of all on initializing the cloud will take place, cloudlet will hand over the task to broker and broker will then submit this task to datacenter to perform where data center will allocate the requirements of that task like allocating VM, bandwidth, RAM etc. There is a host inside datacenter that has VMs inside it. These VMs will perform the tasks submitted by the broker. After completion all these entities datacenter, broker shut down. This can also be shown in Figure 13.

```

Output - Gunjan (run)
run:
Starting Cloud
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
400.1: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID  STATUS  Data center ID  VM ID  Time  Start Time  Finish Time
      0      SUCCESS      2          0    435      0.1      435.1

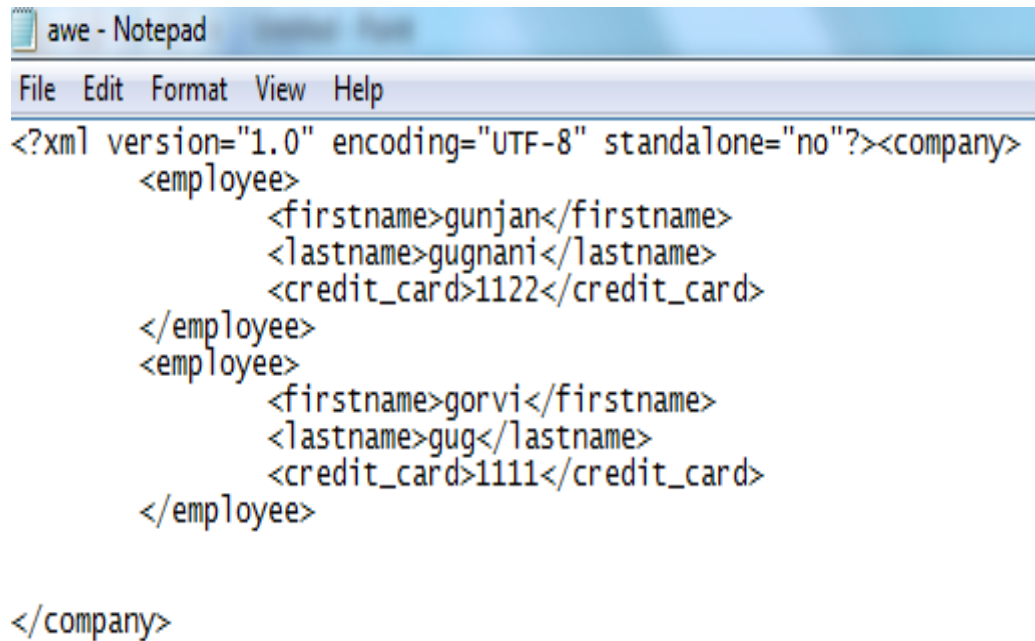
STEP1: READING XML FILE
Root element company
Information of all employees

```

**Figure13: Output of Cloud**

## 4.2 Results

Here, section consists of the analysis of the proposed scheme with the other traditional techniques present in the literature e.g. DES AES, and RSA. The experiments are performed on an XML file. The sample XML file taken for experiment is shown below in the Figure14.



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><company>
  <employee>
    <firstname>gunjan</firstname>
    <lastname>gugnani</lastname>
    <credit_card>1122</credit_card>
  </employee>
  <employee>
    <firstname>gorvi</firstname>
    <lastname>gug</lastname>
    <credit_card>1111</credit_card>
  </employee>
</company>
```

Figure14: XML File



Figure15: GUI for Encryption

First of all a GUI has been created to demonstrate the encryption in cloud as shown in Figure 15. This GUI contains a file path to choose file for encryption along with some buttons of AES, DES, RSA , Attack and Simulate is for DNA encryption.

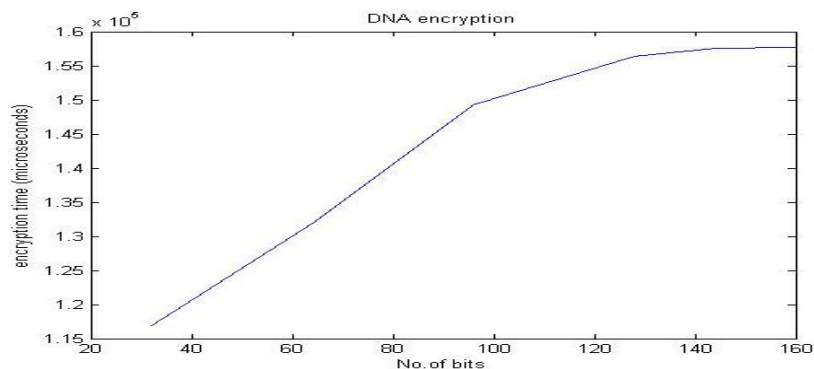
## Assumptions

Some of the assumptions have been taken before performing the experiments.

- a) Credit card pin is assumed to be important field among the other two fields- first name and second name. So as to demonstrate the effect of selective encryption.
- b) During encryption keys are distributed through secure channel.
- c) As injection attack are possible due to less input validations, so to demonstrate the effect of XPath injection attack, deliberately no validation are applied in the code.

## 4.3 Analysis of Proposed Approach

The proposed algorithm DNA XML Encryption (Figure 17) is tested for text length of integer numbers from a 4-digit i.e. 32 message length to 20-digit i.e.140 bit message length. The execution time is in microseconds. The execution time depends on the length of selected text being encrypted. The graph shown in Figure16 represents number of bits and average encryption time. The average encryption time is summation of time taken to encrypt the selected text divided by the total number of time text being encrypted. During the experiment it is observed that as the number of bits increases i.e. if the length of the text increases then the encryption time also increases.



**Figure16: Number of Bits vs. AverageDNAEncryption Time**



```

===== OUTPUT =====
Cloudlet ID   STATUS   Data center ID   VM ID   Time   Start Time   Finish Time
    0         SUCCESS      2             0       425       0.1         425.1

STEP1: READING XML FILE
Root element company
Information of all employees
First Name : gunjan
Last Name  : gugnani
Credit Card : 1234

STEP2: EXTRACTING TEXT FROM XML TO ENCRYPT
string is 1234

STEP3: CONVERTING STRING TO BINARY
'1234' to binary: 00110001001100100011001100110100
result=====00110001001100100011001100110100
string is 201420182020209

STEP7: Replacing ENCRYPTED TEXT IN XML FILE

ENCRYPTION Done

132314
microsec
step1:Reading XML file
Root element company
Information of all employees
First Name : gunjan
Last Name  : gugnani
Encrypted Credit Card detail : 201420182020209
Decrypted Credit Card detail : 1234

```

**Figure17: Output DNA encryption.**

## 4.4 DES Analysis

Again in the DES Encryption as shown in Figure18, the average encryption time is computed for integer as well as characters ranging from a 4-digit i.e. 32 message length to 20-digit i.e.140 bit message length. The execution time  $t$  is in microseconds. The execution time depends on the length of selected text being encrypted and it is also been observed that the change in graph depends on the value of plain text, that is the average time taken to encrypt increases when there is a use of combination of text and digits. This is the reason why there is an abrupt change after 128 bits because at the time of experiment the text enter after 128 bits was the text/string. Figure19 represents the graph between the numbers of bits encrypted vs. average encryption time of DES

```

StartPage  DNA.java  Output - cloud_full (run)  DES.java
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
400.1: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID  STATUS  Data center ID  VM ID  Time  Start Time  Finish Time
0           SUCCESS  2              0      430   0.1         430.1

STEP1: READING XML FILE
Root element company
Information of all employees
First Name : gunjan
Last Name  : gugnani
Credit Card : iDPo6Ebu//MEcN7+hmm/9Q==

STEP2: EXTRACTING TEXT FROM XML TO ENCRYPT
string is iDPo6Ebu//MEcN7+hmm/9Q==
clear message: iDPo6Ebu//MEcN7+hmm/9Q==
encrypted message: 9xD1c7+ArJhSKalmUgD22eBDq20BedbVxWBoUsst4ro=
STEP7: Replacing ENCRYPTED TEXT IN XML FILE

ENCRYPTION Done
encryption time is216025

decrypted message: iDPo6Ebu//MEcN7+hmm/9Q==

cloud_full (run)  running...  182 | 29 | INS

```

Figure 18: Output DES

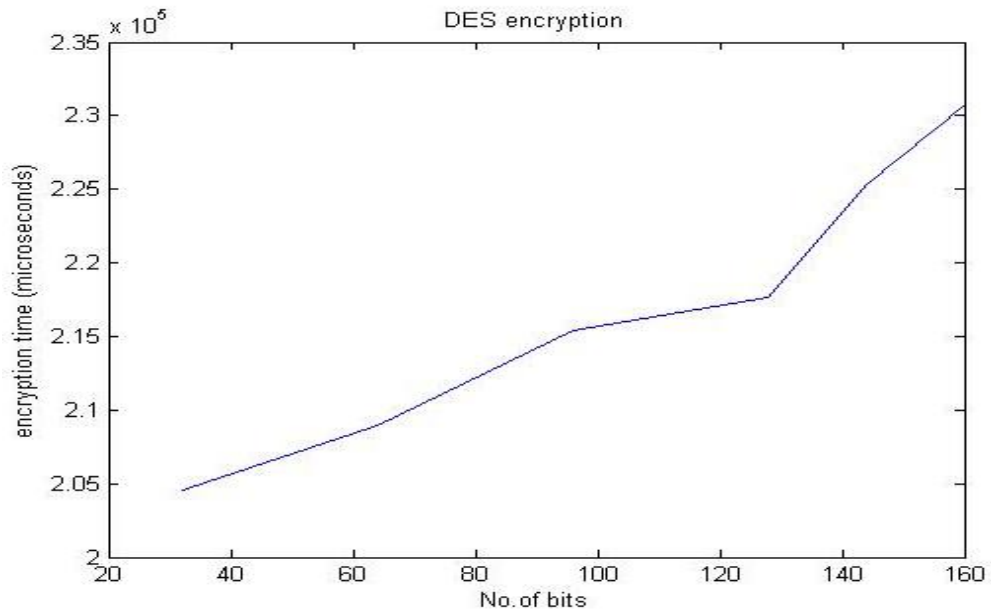


Figure19: Average Encryption Time of DES vs. No. of Bits

## 4.5 AES Analysis

AES is again used here to provide confidentiality in cloud environment. XML selected field is encrypted using AES. Here it is observed that average encryption time taken by AES is significantly high. Here one more point is observed that the average encryption time taken by AES increases if there is a use of text and digits collectively. Here in figure20: we can see the output of AES encryption and decryption also in the Figure21: represents the graph between the number of bits encrypted vs. average encryption time of AES.

```
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID   STATUS   Data center ID   VM ID   Time   Start Time   Finish Time
    0         SUCCESS       2           0     432       0.1         432.1
STEP1: READING XML FILE
Root element company
Information of all employees
First Name : gunjan
Last Name : gugnani
Credit Card : 201420182020209

STEP2: EXTRACTING TEXT FROM XML TO ENCRYPT
string is 201420182020209
string is iDPo6Ebu//MEcN7+hmw/9Q==

STEP7: Replacing ENCRYPTED TEXT IN XML FILE
iDPo6Ebu//MEcN7+hmw/9Q==

ENCRYPTION Done
encryption time =316357

microsec

201420182020209
```

Figure20: Output of AES

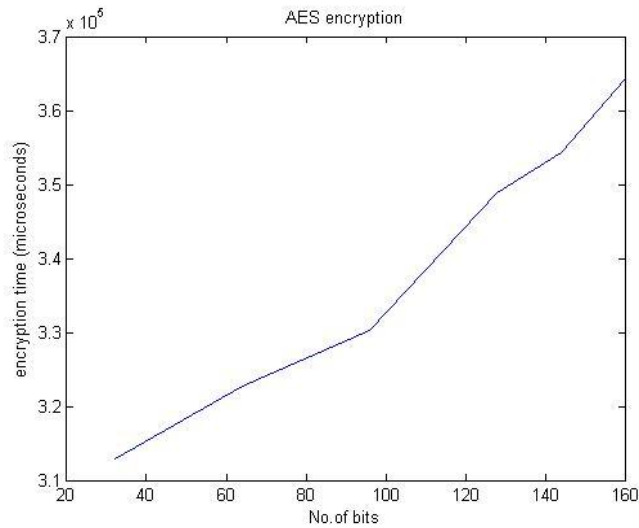


Figure21: Average Encryption Time of AES vs. No. of Bits

## 4.6 RSA Analysis

```

Start Page  DNA.java  Output  DES.java  RSA.java
cloud_full (run)  cloud_full (run) #2
0.1: Broker: Sending cloudlet 0 to VM #0
400.1: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID  STATUS  Data center ID  VM ID  Time  Start Time  Finish
0           SUCCESS  2              0      498  0.1         498.1
STEP1: READING XML FILE
Root element company
Information of all employees
First Name : gunjan
Last Name  : gugnani
Credit Card : 53471

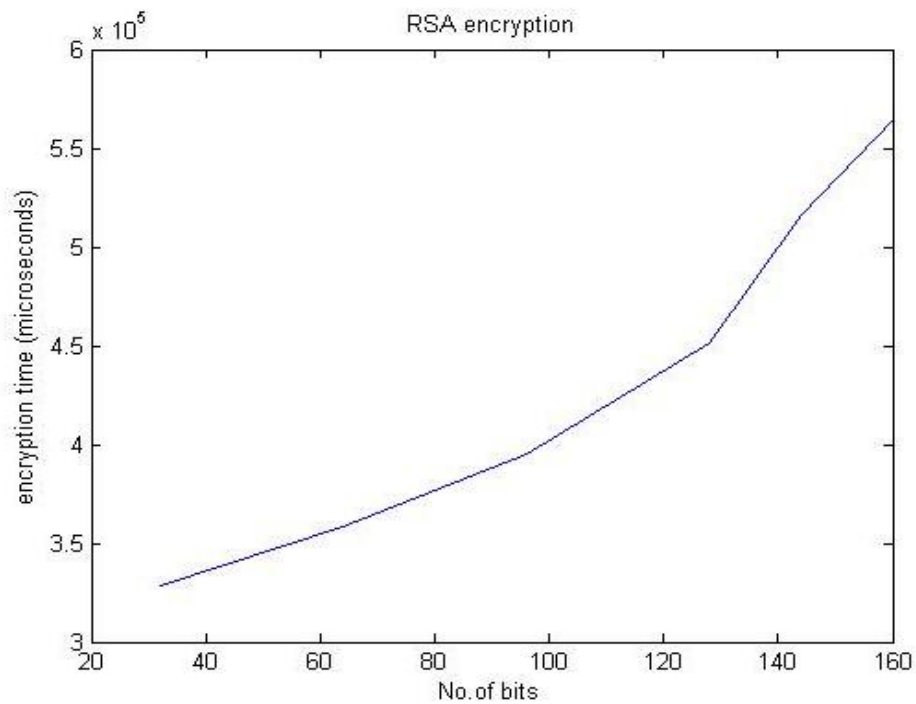
STEP2: EXTRACTING TEXT FROM XML TO ENCRYPT
string is 53471
Plaintext: 53471
Ciphertext: 11929297245370824146204105080852625
STEP7: Replacing ENCRYPTED TEXT IN XML FILE

ENCRYPTION Done

encryption time =562937
microsec
Plaintext: 53471
  
```

Figure22: Output of RSA

As in all the other cases RSA Encryption as shown in Figure22 is tested for integer numbers from a 4- digit i.e. 32 message length to 20-digit i.e. 140 bit message length. The execution time  $t$  is in microseconds. The execution time depends on the length of selected text being encrypted. This experiment is performed with the values of  $p=2$  and  $q=3$  as prime numbers. With  $p=2$  and  $q=3$  it is observed that the average encryption time of RSA increases with the increase in text length. Figure23 represents the graph between the numbers of bits encrypted vs. average encryption time.

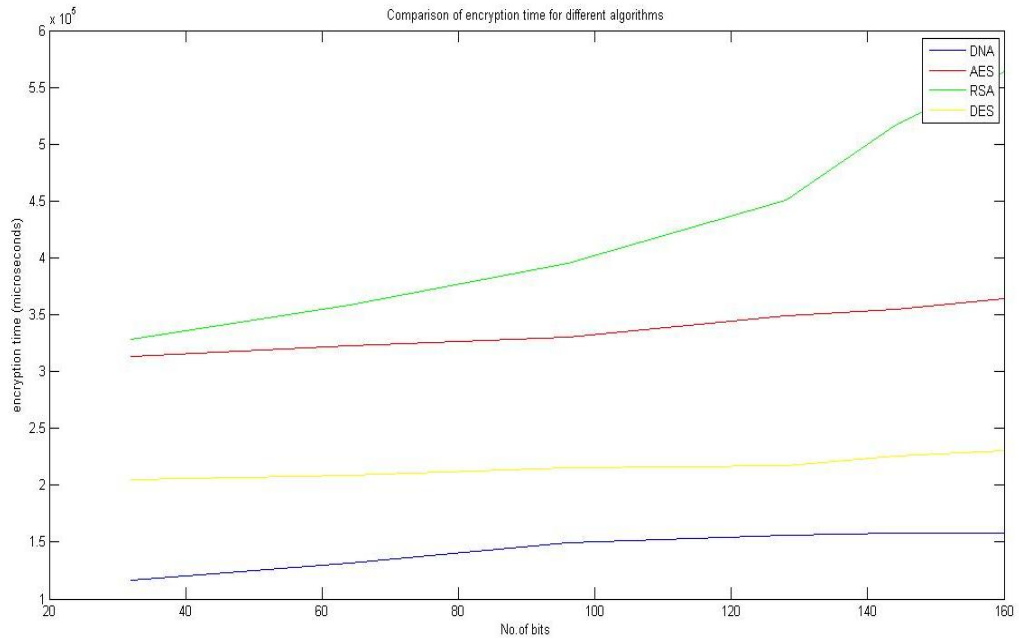


**Figure23: Average Encryption time of RSA vs. No. of Bits**

## 4.7 Comparison with Traditional Schemes

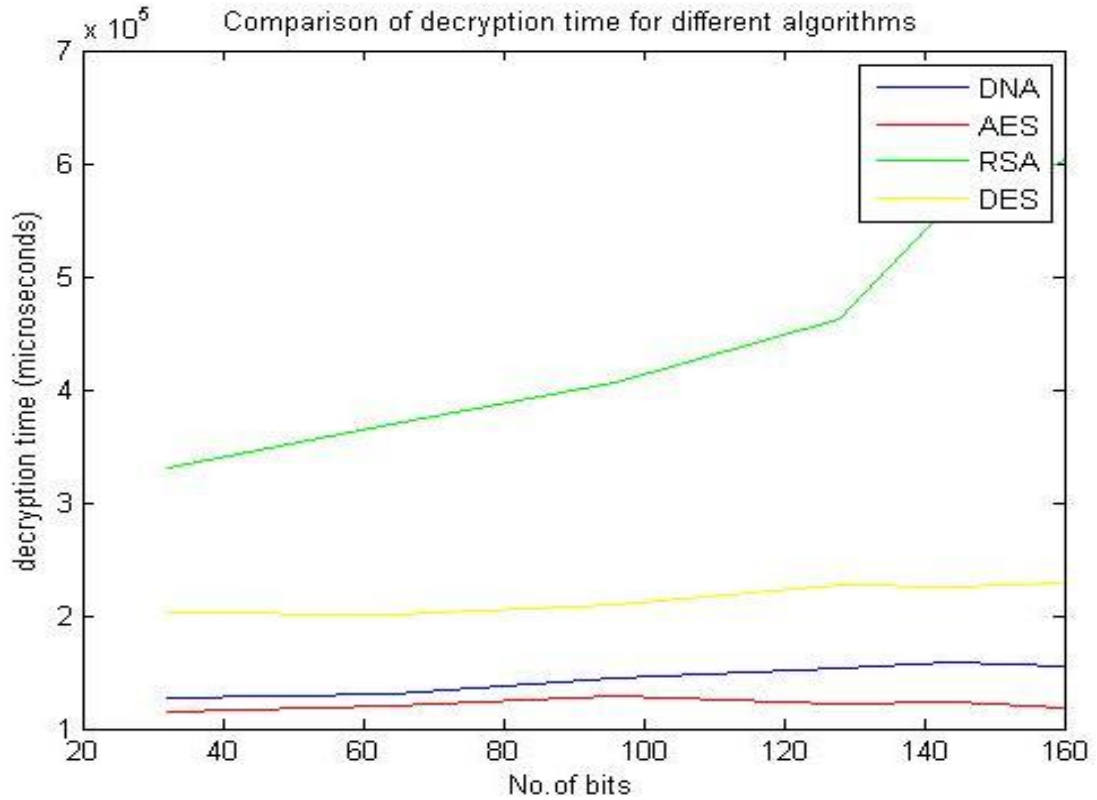
The figure24: represents the graph between the number of bits encrypted vs. average encryption time of DNA, AES, RSA and DES. It has been observed that the average time taken by the DNA encryption is low than the other three followed by DES, AES and the maximum average encryption time is for RSA. Moreover, it is also know that DNA cipher is a strong cipher. Now if the attacker wishes to know about the credit card pin of the person then he needs to decrypt it first. As mentioned earlier the probability to decrypt it with a correct guess is .0000000061 and this can be further reduced because we have four DNA bases A,C,T, and G , now assigning them binary digits as A=00, C=01, T=10 and G=11 can take  $4!$  combinations. Moreover in the

complementary rule we have A=C, G=T, T=A, C=G, this can also take 4! combinations. This will result as  $.0000000061/(4!*4!)$ , which gives us the probability as  $1.065098841*10^{-11}$ . From the lowest encryption time and low probability to guess the right text, we came to the conclusion that XML DNA encryption is appropriate for improving not only the security and confidentiality of the cloud but also improves the performance of the cloud by lowering the encryption time.



**Figure24: Comparison between DNA, AES, DES and RSA**

If we consider the decryption in account then lots of points are there to be noticed. In case of DNA decryption, there is very minute difference between DNA encryption and decryption time. In case of DES, the decryption time is very less as compare to the encryption time. In case of AES, there is again very minute difference between encryption and decryption time and in every run of AES the decryption time comes a little less than encryption time. In case of RSA the decryption time is always greater than encryption time. Hence the average decryption time of RSA is always greater than the average encryption time of RSA. All of these observations are shown in Figure25.



**Figure25: Average Decryption Time for Different Algorithms**

#### 4.8 Comparison with Every Field Encryption

Earlier in all the experiments the comparison is made while encrypting the single field in the XML file. In the following experiments the encryption is made while encrypting all the fields of XML files. Again in this experiment we have computed the average encryption time which is here the summation of total time taken by the algorithm to encrypt all the three fields by the number of time the experiment is performed. During the experiment it is observed that while encrypting all the three fields the time taken by the algorithm is significantly very high as compared to the time taken by the algorithm to encrypt the single field. This shows that its better to selectively encrypt only those elements that are confidential, in this manner the we can prevent a cloud user for improper discloser or leakage of the information and moreover the encryption time is also reduced in selective encryption consequently the performance of the cloud will increase.

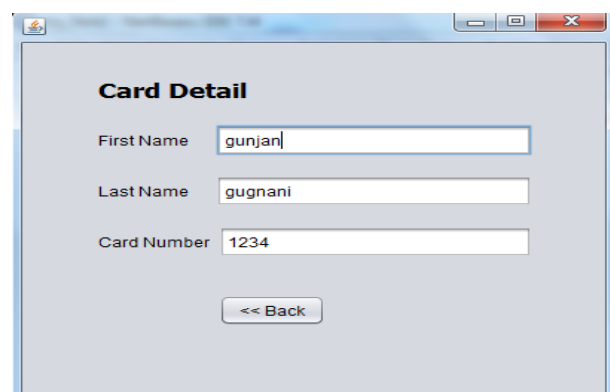
## 4.9 Attack

To demonstrate the benefit of encryption, An XPath injection attack is performed on the XML file. This XPath injection is performed when no proper input validations are used inside the code. Attacker takes advantage of this vulnerability and breaks the confidentiality of the user information so as a solution it is suggested that to keep the data in encrypted form in the file. The DNA encryption is performed to encrypt the fields of XML file. To demonstrate the attack a GUI is created to such that the user can enter the details and when user enters the, even if he doesn't know both the last name and first name, the attacker be able to fetch the card details. But we can prevent this attack by just storing the data in encrypted form. The attack is demonstrated in the Figure 26 and Figure 27.



**Figure26: GUI for Attack**

As no proper input validations are applied in code, if the attacker fill the first name and any random last name (because no input validations are there) so when he will perform the attack he will get to know the card details as shown in Figure 27.



**Figure27: Fetched details via Attack**



# Chapter 5

**Conclusion**

**Future Scope**

# **CHAPTER - 5**

## **CONCLUSION**

### **5.1 Conclusion**

From this work we can conclude that to improve the security, confidentiality, privacy of a cloud user, we must store the data in encrypted form but if we will encrypt the whole data from very minute user information to confidential information then it will raise a performance issue both in the terms of encryption time and storage as they length of encrypted text is always greater than the plain text. Therefore there is a strong need to maintain a balance with the confidentiality and the performance of the system. So as a solution to maintain balance among confidentiality along with the performance we can go for selectively encryption that is we will encrypt only the confidential information and leave can leave the other information as plain text. In this work we achieve this balance through selective DNA encryption and demonstrate how it selectively encrypting confidential information and then storing in encrypted form and thus prevent it from various kind of information discloser and information leakage. In this work we also demonstrate how this storing data in encryption form prevents data from information discloser through the XPath attack.

### **5.2 Future Work**

In this work we are just implementing the confidentiality and are maintaining a balance within confidentiality and performance issue. But along with the confidentiality, integrity is also a major aspect. So in terms to embed integrity along with the confidentiality we can do for XML signature. XML signature provides integrity to the user information so that nobody will able to tamper with the user information. With both the XML encryption and XML signature both the confidentiality and integrity will be achieved which is making the whole system highly secure.

## **List of Publications**

1. Gugnani et al. "Implementing DNA Encryption Technique in Web Services to Embed Confidentiality in Cloud," 2nd International Conference on Computer and Communication Technologies, Springer, Hyderabad, India, 2015.  
*(Accepted)*
2. P. K. Gupta, GunjanGugnani, S.P. Ghrrera, "XML DNA Encryption: Improve Security of Cloud Applications," CSI Communications, 2015.  
*(Communicated)*

## References

- [1] P.Mell and T.Grance, "The NIST Definition of Cloud Computing," US National Institute of Standards and Technology, 2011; <http://www.nist.gov/itl/csd/cloud-102511.cfm> (Accessed 25 Sept 2014)
- [2] Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Avi Patel, and Muttukrishnan Rajarajan, "A survey on security issues and solutions at different layers of Cloud computing," *Journal of Supercomputing*, vol. 63, issue 2, pp.561-592, 2013.
- [3] Kandukuri BR, Paturi VR, and Rakshit, A, "Cloud security issues," *IEEE international conference on services computing*, Bangalore, pp. 517-520, 2009.
- [4] Yau, S., S., and Ho G., "Protection of users' data confidentiality in cloud computing," *In Proceedings of the 2nd Asia-Pacific Symposium on Internetware*, New York, NY, pp. 1-6, 2010.
- [5] Richa H. Ranalkar, and B.D. Phulpagar, "DNA based Cryptography in Multi-Cloud: Security Strategy and Analysis," *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)* vol. 3, Issue 2, pp.189-192, 2014
- [6] Mohammad Reza Abbasy, and Bharanidharan Shanmugam, "Enabling Data Hiding for Resource Sharing in Cloud Computing Environments Based on DNA Sequences" *IEEE World Congress on Services*, Washington, DC, pp.385-390, July 2011.
- [7] Anandita Thakur, and P.K Gupta, "Framework to Improve Data Integrity in Multi Cloud," *International Journal of Computer Applications*, vol. 87, issue 10, pp.28-32, 2013.
- [8] Yuyu Chou, Olga Levina, and Jan Oetting, "Enforcing Confidentiality in a SaaS Cloud Environment," *IEEE 19th Telecommunications forum*, pp.90-93, November 2011..
- [9] J.Srinivas, K.VenkataSubba Reddy, and A.Moiz Qyser, "Cloud Computing Basics," *International Journal of Advance Research in Computer and Communication Engineering*, vol. 1, no. 5, pp. 343-347, 2012.
- [10] Vaquero, Luis M., Luis Rodero-Merino, and Daniel Morán, "Locking the sky: a survey on IaaS cloud security," *Computing*, vol.91, no.1, pp.93-118, 2011.
- [11] M.G. Jaatun, G.Zhao and C.Rong, "Strengthen Cloud Computing Security with Federal Identity Management Using Hierarchical Identity-Based Cryptography," *Cloud Computing, Springer Berlin, Heidelberg*, pp.167-177, 2009

- [12] GuYue-sheng, Ye Meng-tao, and Y Gan, "Web Services Security Based on XML Signature and XML Encryption," *Journal of Networks*, vol. 5, no. 9, pp. 1092-1097, 2010
- [13] Bernd Grobauer, Tobias Walloschek, and Elmar Siemens, "Understanding Cloud Computing Vulnerabilities," *IEEE Security and Privacy*, vol. 9, no. 2, pp.50-57, 2011.
- [14] Herzog, P., "Open Source Security Testing Methodology Manual (OSSTMM)," *Institute for Security and Open Methodologies (ISECOM)*, 2003. [Online] <http://www.isecom.org/research/osstmm.html> (Accessed 12 Sept 2014)
- [15] Krutz, Ronald L., and Russell Dean Vines, "Cloud Security: A Comprehensive Guide to Secure Cloud Computing," John Wiley & Sons, 2010.
- [16] Subashini, S. and Kavitha, V, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol.34, no.1, pp.1-11, 2011.
- [17] Te-Shun Chou, "Security Threats On Cloud Computing Vulnerabilities," *International Journal of Computer Science & Information Technology (IJCSIT)*, vol. 5, no 3, pp. 79-88, June 2013
- [18] Liu, Hongjun, Da Lin, and AbdurahmanKadir, "A novel data hiding method based on deoxyribonucleic acid coding." *Computers & Electrical Engineering*, vol. 39, no.4, pp.1164-1173, 2013.
- [19] Terec, R., Vaida, M. F., Alboaie, L., & Chiorean, L., "DNA security using symmetric and asymmetric cryptography," *International Journal of New Computer Architectures and Their Applications (IJNCAA)*, vol.1, no.1, pp.34-51, 2011.
- [20] Lee, Ken Ka-Yin, Wai-Choi Tang, and Kup-Sze Choi, "Alternatives to relational database: comparison of NoSQL and XML approaches for clinical data storage," *Computer methods and programs in biomedicine*, vol.110 no.1 pp.99-109, 2013.
- [21] Shiu, H. J., Ng, K. L., Fang, J. F., Lee, R. C., & Huang, C. H., "Data hiding methods based upon DNA sequences," *Information Sciences*, vol.180, no.11, pp.2196-2208, 2010.
- [22] Moradian, Esmiralda, and Anne Håkansson, "Possible attacks on XML web services," *Int. J. Computer Science and Network Security (IJCSNS)*, vol.6, no.1, pp.154-170, 2006.
- [23] M.Jensen, J.O.Schwenk, N.Gruschka, and L.L.Iacono, "On Technical Security," *IEEE International Conference on Cloud Computing (Cloud-II 2009)*, Bangalore, India, 2009, pp.109-116.

- [24] DuanYouxiang, Gao Yang, "Evaluating Vulnerabilities Quantitatively Based On the Rank of Web Services Confidentiality," *Journal of Next Generation Information Technology (JNIT)*, vol.2, no. 1, pp. 81-87, 2011.
- [25] Md. Syed Mahamud Hossein, "A Compression and Encryption Algorithms on DNA Sequences using R<sup>2</sup>CP and modified Huffman Technique," *International Journal of Computer Applications*, vol. 57 no.1, pp.1-10, 2012.
- [26] Anam, Beenish, Kazi Sakib, Md Hossain, and Keshav Dahal, "Review on the Advancements of DNA Cryptography," *arXiv preprint*, pp.1-7, 2010, <http://arxiv.org/pdf/1010.0186.pdf> (Accessed 10 March, 2015)
- [27] Yasha Parthi, and Sunita Dixit, "A Comparative Study on DNA Cryptography," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, Issue5, May 2014.
- [28] X. Zhang, and K. K. Parhi, "High-speed VLSI architectures for the AES algorithm," *IEEE Trans. VLSI Systems*, vol.12, Issue.9, pp. 957 - 967, 2004.
- [29] Saqib, Nazar A., Francisco Rodriguez-Henriquez, and Arturo Diaz-Pérez, "A compact and efficient FPGA implementation of the DES algorithm," In *International Conference on Reconfigurable Computing and FPGAs*, pp. 12-18, 2004.
- [30] T. Beth, and D. Gollmann, "Algorithm engineering for public key algorithm," *IEEE Journal on Selected Areas in Communications*, vol.7, no.4, pp.458-465, May 1989.
- [31] Fei Hu, Meikang Qiu, Jiayin Li, Travis Grant, Draw Tylor, Seth McCaleb, Lee Butler and Richard Hamner, "A Review on Cloud Computing: Design Challenges in Architecture and Security," *Journal of Computing and Information Technology – CIT*, vol.19, no.1, pp.25–55, 2011.
- [32] CloudSim: A Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services [Online] <http://www.cloudbus.org/cloudsim/> (Accessed 9 May 2015)
- [33] Pradeep Bhosale, Priyanka Deshmukh, Girish Dimbar, and Ashwini Deshpande, "A Review Paper on Enhancing Data Security in Cloud Computing Using 3D Framework & Digital Signature with Encryption," *International Journal of Engineering Research & Technology (IJERT)*, vol. 1, Issue 8, pp.1-8, 2012.
- [34] Kevin Hamlen, Murat Kantarcioglu, Latifur Khan, and Bhavani Thuraisingham, "Security Issues for Cloud Computing," *International Journal of Information Security and Privacy*, vol.4, no.2, pp.39-51, 2010.
- [35] Huaglory Tianfield, "Security Issues in Cloud Computing," *IEEE International Conference on Systems, Man, and Cybernetics (SMC'12)*, Seoul, Korea, pp. 1082-1089, 2012.

- [36] Deyan Chen and Hong Zhao, "Data Security and Privacy Protection Issues in Cloud Computing," *International Conference on Computer Science and Electronics Engineering, Hangzhou* , pp.647-651, March 2012.

## *Appendix*

### *DNA encryption*

```
try {

    System.out.println("STEP1: READING XML FILE ");
    String filepath = "F:\\attack\\awe.xml";
    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
    DocumentBuilder db = dbf.newDocumentBuilder();
    Document doc = db.parse(filepath);
    doc.getDocumentElement().normalize();

    System.out.println("Root element " + doc.getDocumentElement().getNodeName());

    NodeList nodeList = doc.getElementsByTagName("employee");
    System.out.println("Information of all employees");

    for (int s = 0; s <1; s++) {

        Node fstNode = nodeList.item(s);
        NodeList fstNm=null;
        Element fstNmElmnt=null;
        NodeList fstcredit=null;

        if (fstNode.getNodeType() == Node.ELEMENT_NODE) {

            Element fstElmnt = (Element) fstNode;
            NodeList fstNmElmntLst = fstElmnt.getElementsByTagName("firstname");
            fstNmElmnt = (Element) fstNmElmntLst.item(0);
            fstNm = fstNmElmnt.getChildNodes();
            System.out.println("First Name : " + ((Node) fstNm.item(0)).getNodeValue());
            NodeList lstNmElmntLst = fstElmnt.getElementsByTagName("lastname");
            Element lstNmElmnt = (Element) lstNmElmntLst.item(0);
            NodeList lstNm = lstNmElmnt.getChildNodes();
            System.out.println("Last Name : " + ((Node) lstNm.item(0)).getNodeValue());
            NodeList creditLst = fstElmnt.getElementsByTagName("credit_card");
            Element creditElmnt = (Element) creditLst.item(0);
            fstcredit = creditElmnt.getChildNodes();
            System.out.println("Credit Card : " + ((Node) fstcredit.item(0)).getNodeValue());
        }

        System.out.println("");
    }
}
```



```

System.out.println("STEP2: EXTRACTING TEXT FROM XML TO ENCRYPT");
str =((Node) fstcredit.item(0)).getNodeValue() ;
System.out.println("string is " + str );
byte[] bytes = str.getBytes();
StringBuilder binary1 = new StringBuilder();
for (byte b : bytes)
{
    int val = b;
    for (int i = 0; i < 8; i++)
    {
        binary1.append((val & 128) == 0 ? 0 : 1);
        val <<= 1;
    }
    // binary1.append(' ');
}
System.out.println("");

```

```

System.out.println("STEP3: CONVERTING STRING TO BINARY");
System.out.println(""" + str + "" to binary: " + binary1);

```

```

String result = binary1.toString();
System.out.println("result=====" + result);

```

```

String strn=result;
char[] chars=strn.toCharArray();

```

```

for (int i = 0; i < chars.length; i++) {

```

```

    String str4 = "" + chars[i];
    i++;
    String str5= "" + chars[i];

```

```

    str4 =str4.concat(str5);
    System.out.println(str4);
    str6=str6.concat(str4);
    count++;
}

```

```

String[] scripts = new String[count];
for (int i = 0; i < chars.length; i++) {
    String str4 = "" + chars[i];
    i++;
    String str5= "" + chars[i];

```

```

        str4 =str4.concat(str5);
scripts[j]=str4;
        j++;
    }

```

```

System.out.println("string=====" + str6);
System.out.println("array");
    for ( j = 0; j < scripts.length; j++) {
        System.out.print(scripts[j]);
    }
    System.out.println("");

```

```

System.out.println("STEP4:ASSIGING BINARY TO DNA BASES");
String[] scripts1 = new String[count];
    String[] strarr = new String[count];
for ( j = 0; j < scripts.length; j++) {

    str7=scripts[j];

    if (str7.equals("00")) {

        str7="A";

        strarr[j]=str7;
        System.out.print(strarr[j]);
    }
else if (str7.equals("01")) {

    str7="T";
    strarr[j]=str7;
    System.out.print(strarr[j]);
    }
else if (str7.equals("10"))

    str7="C";
    strarr[j]=str7;
    System.out.print(strarr[j]);
    }
else {

    str7="C";
    strarr[j]="G";
    System.out.print(strarr[j]);
    }
}

```

```

}
System.out.println("");

System.out.println("STEP5:After replacing DNA complements");
String[] strarr1 = new String[count];
for ( j = 0; j < scripts.length; j++)
{
    str9=strarr[j];

    if (str9.equals("A"))
);

    str9="C";

    strarr1[j]=str9;
    System.out.print(strarr1[j]);}
    else if(str9.equals("G")) {

    str9="T";

    strarr1[j]=str9;
    System.out.print(strarr1[j]);
    }
    else if(str9.equals("T")) {

    str9="A";

    strarr1[j]=str9;
    System.out.print(strarr1[j]);
    }
    else{

    str9="G";

    strarr1[j]=str9;
    System.out.print(strarr1[j]);
    }
    }
System.out.println("");
int inc=0;

for ( j = 0; j < strarr1.length; j++){

    strl3=strarr1[j];
    j++;
    strl31=strarr1[j];

```

```

    strarrl3[inc]=strl3.concat(strl31);
    inc++;
System.out.println("");
System.out.println("arrayfinal");
System.out.println(Arrays.toString(strarrl3));
String[][] reference = new String[20][2] ;
reference[0][0] = "AA";
reference[0][1] = "1";
reference[1][0] = "AC";
reference[1][1] = "2";
reference[2][0] = "AT" ;
reference[2][1] = "3";
reference[3][0] = "AG";
reference[3][1] = "4";
reference[4][0] = "CC";
reference[4][1] = "5";
reference[5][0] = "CA";
reference[5][1] = "6";
reference[6][0] = "CT" ;
reference[6][1] = "7";
reference[7][0] = "CG";
reference[7][1] = "8";
reference[8][0] = "TT";
reference[8][1] = "9";
reference[9][0] = "TA";
reference[9][1] = "10";
reference[10][0] = "TC" ;
reference[10][1] = "11";
reference[11][0] = "TG";
reference[11][1] = "12";
reference[12][0] = "GG";
reference[12][1] = "13";
reference[13][0] = "GA";
reference[13][1] = "14";
reference[14][0] = "GC" ;
reference[14][1] = "15";
reference[15][0] = "GT";
reference[15][1] = "16";
reference[16][0] = "TC";
reference[16][1] = "17";
reference[17][0] = "CG" ;
reference[17][1] = "18";
reference[18][0] = "TG";
reference[18][1] = "19";
reference[19][0] = "CT" ;
reference[19][1] = "20";
System.out.println(Arrays.deepToString(reference));
String[] scripts2 = new String[count/2];
for(int i=0;i<strarrl3.length;i++)
{

```

```

for(int t=0;t<reference.length;t++)
{

if((strarrl3[i].equals(reference[t][0]))
{
    scripts2[i] = reference[t][1];

}
}

}

System.out.println("");

System.out.println("STEP6: USING DNA REFERENCE SEQUENCE
ENCRYPTING BASES WITH THEIR CORRESPONDING NUMBERS ");
System.out.println(Arrays.toString(scripts2));

String final1,final2,final3="";
for(int i=0;i<scripts2.length;i++)
{
    final1=scripts2[i];

    final3=final3.concat(final1);
}
    System.out.println("string is "+final3);
Node staff = doc.getElementsByTagName("employee").item(0);
NodeList list = staff.getChildNodes();
for (int i = 0; i < list.getLength(); i++) {
    Node node = list.item(i);
    if ("credit_card".equals(node.getNodeName())) {

        //node.setTextContent(final3);
        node.setTextContent(final3);
    }
}
//}
System.out.println("");

System.out.println("STEP7: Replacing ENCRYPTED TEXT IN XML FILE");
    TransformerFactory transformerFactory =
        TransformerFactory.newInstance();
    Transformer transformer = transformerFactory.newTransformer();
    DOMSource source = new DOMSource(doc);
    StreamResult result1 = new StreamResult(new File(filepath));
    transformer.transform(source, result1);
    System.out.println("");

```

```
System.out.println("ENCRYPTION Done");

    long t2=System.nanoTime();
    System.out.println("");
long t3=((t2-t1)/1000);
System.out.println("encryption time" + t3);

System.out.println("microsec");
}
```